Università degli Studi di Padova

FACOLTÀ DI INGEGNERIA

Corso di Laurea Specialistica in Ingegneria delle Telecomunicazioni

Tesi di Laurea

Codifica di immagini multi-view con trasformata Wavelet

Relatore: Prof. P. Zanuttigh

Laureando: Andrea Sandonà

5 ottobre 2010

Sommario

La codifica di immagini e video multiview rappresenta un ambito di ricerca di ampio interesse sia da parte del mondo accademico che da quello industriale. In un insieme di immagini multiview la quantità di informazione da gestire è molto elevata, diventa perciò indispensabile lo studio di tecniche di compressione che siano efficienti e allo stesso tempo garantiscano una buona qualità visiva delle sequenze ricostruite. L'approccio classico per la codifica di dati multiview si basa sostanzialmente su estensioni delle tecniche di compressione di immagini e video: tipicamente la compensazione del moto che viene applicata lungo la dimensione temporale viene applicata anche attraverso le varie viste. La trasformata wavelet è uno strumento molto utilizzato nella codifica delle immagini. Lo standard JPEG2000, ad esempio, il quale è stato introdotto per sostituire il classico standard JPEG, fa uso della trasformata wavelet al posto della DCT, rendendo la qualità dell'immagine sicuramente migliore. In questo lavoro di tesi vengono proposti due approcci di codifica di immagini multiview basati su trasformata wavelet. In entrambi gli algoritmi implementati si cerca di ridurre la ridondanza tra le viste sfruttando i dati geometrici della scena con operazioni di proiezione (Warping) e attraverso una processo noto come *wavelet lifting*. Successivamente si comprimono i coefficienti ottenuti con un codificatore JPEG2000. La differenza sostanziale tra i due algoritmi sta nel modo in cui viene applicato il warping. Nel primo caso infatti tale operazione viene applicata prima del wavelet lifting, rispetto a una singola vista di riferimento e gestendo le occlusioni in modo indipendente. Nel secondo caso invece il warping viene implementato all'interno della wavelet tra le viste adiacenti. Dai dati sperimentali si è potuto constatare che il primo approccio risulta essere più efficiente soprattutto a bassi bit rate mentre per quanto riguarda il secondo approccio risulta necessaria una codifica dei coefficienti ad hoc.

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Introduzione

La codifica di immagini e video multiview, grazie al miglioramento delle tecnologie di elaborazione e visualizzazione delle immagini e alle innovative applicazioni derivanti da questa tecnica, è motivo di ricerca sempre maggiore sia da parte del mondo accademico che da quello industriale. Le applicazioni che si possono affrontare, infatti, con questo tipo di tecnologia sono molteplici e sono sostanzialmente applicazioni di tipo multimediale e iterativo. In generale questi tipi di applicazioni possono essere raggruppate sotto tre grandi categorie:

- Free-viewpoit video (FVV)
- Three-dimensional television (3DTV)
- Immersive teleconferencing.

Un insieme di immagini multiview viene solitamente generato attraverso un set di videocamere che da differenti punti di vista riprendono lo stesso oggetto o scena. La quantità di informazione da gestire risulta quindi molto elevata, e perciò è necessaria una compressione efficiente dei dati, garantendo tuttavia una buona qualità visiva della sequenza ricostruita.

Negli ultimi dieci anni, sono state proposte diverse tecniche di multiview video coding (MVC). Il punto chiave di queste tecniche sta nell' utilizzare in

modo efficiente la correlazione tra le viste adiacenti, oltre che le correlazioni temporali e spaziali tra frame successivi, già ampiamente studiate nelle codifiche video tradizionali.

Uno degli strumenti più performanti che viene utilizzato nella codifica delle immagini digitali è sicuramente la *trasformata wavelet*. Questo operatore consente di ottenere un'insieme di coefficienti che garantiscono un elevata efficienza di compressione. La trasformata wavelet è stata utilizzata ad esempio nello standard JPEG2000, nato come alternativa allo standard JPEG (il quale fa uso di DCT), rendendo la codifica molto più performante soprattutto a bassi bit rate.

In questo lavoro sono stati implementati e analizzati due algoritmi dove la trasformata wavelet viene applicata alle immagini multiview. In entrambi i casi si cerca di eliminare la ridondanza esistente tra le viste attraverso l'applicazione della trasformata wavelet e delle informazioni geometriche della scena e successivamente i coefficienti ottenuti vengono compressi con JPEG2000.

Di seguito viene riportata l'organizzazione della tesi.

Nel secondo capitolo vengono introdotti i concetti base. Vengono definite le immagini e i video digitali, le loro estensioni al multiview, tutte le nozioni fondamentali legate alla geometria della scena e i vari modelli adottati. Verranno illustrati inoltre alcuni indici di prestazione utilizzati per valutare gli algoritmi.

Nel terzo capitolo viene fatta un breve trattazione della teoria delle wavelet. Viene descritta la trasformata wavelet continua, come ottenere la trasformata discreta e come le wavelet discrete possono essere implementate attraverso un banco di filtri rendendo molto più semplice e immediata la loro realizzazione. Verrà inoltre descritta come viene utilizzata la trasformata wavelet nelle immagini, e verrà fatta una breve introduzione al JPEG2000.

Nel quarto capitolo, viene analizzato un nuovo tipo di approccio nell'utilizzare le wavelet partendo dall'algoritmo innovativo proposto da Zamarin

et al. in [1] il quale fa uso di 3D-DCT. In questa tesi la 3D-DCT viene sostituita con uno schema di wavelet lifting. L'algoritmo consiste nell'applicare le operazioni di warping prima della trasformata wavelet e gestire le zone occluse in modo indipendente.

Nel capitolo successivo viene presentato il secondo approccio analizzato, il quale rappresenta il classico schema di utilizzo della trasformata wavelet. Lo schema adottato è definito come *wavelet lifting* e in un contesto multiview prende il nome di *Disparity Compensation wavelet lifting* (DCWL) o *Disparity Compensation view filter* (DCVF) dove le operazioni di warping vengono fatte all'interno dello schema.

Nel sesto capitolo vengono riportati i risultati sperimentali dei due algoritmi implementati, analizzando in modo particolare il PSNR medio come indice di prestazione. Nel capitolo sette le conclusioni.

Capitolo 2

Background

In questo capitolo verrà fatta una breve introduzione alle immagini digitali, ai video, e alla loro estensione al multiview. Verranno poi introdotte alcune nozioni di base riguardanti la geometria e la formazione dell'immagine, nozioni indispensabili per poter capire le proiezioni tra le varie viste. Infine varranno mostrati alcuni indici utilizzati per valutare la qualità della codifica.

2.1 Immagini e Video Digitali

Le *immagini e i video digitali* possono essere rappresentati in vari modi. Generalmente le immagini vengono rappresentate da una *matrice bidimensionale* $x[n, m]$ con $0 \leq n < N$, $0 \leq m < M$, dove N e M sono numeri interi finiti e rappresentano rispettivamente l'altezza e la larghezza dell'immagine. Ogni elemento della matrice corrisponde a un campione dell'immagine, il quale viene chiamato "Pixel"(**Pic**ture **el**ement) o "Pel", e la sua posizione all'interno dell'immagine è univocamente determinata dall'indice di riga n e di colonna m.

Le immagini digitali si possono dividere in due categorie: *immagini in scala di grigio* e *immagini a colori*. Nelle immagini in scala di grigio ogni pixel $x[n, m]$ rappresenta l'intensità luminosa (brightness) dell'immagine in

quel punto e solitamente viene espresso attraverso un numero intero, con o senza segno, di B-bit. Nel nostro caso B=8 e i valori sono numeri interi senza segno, per cui $x[n,m] \in \{0,1,...,2^8-1\}, \forall(n,m)$. Nelle immagini a colori ogni pixel viene *tipicamente* rappresentato mediante la tripletta di valori RGB; valori che corrispondono alle tre componenti dei colori primari rosso (R, Red), verde (G, Green), blu (B, Blue). L'immagine viene quindi rappresentata attraverso tre matrici separate $x_c[n,m], c \in \{R,G,B\}$.

Un'importante caratteristica del sistema visivo umano (HVS, *Human Vision System*) consiste nell'avere una diversa sensibilità nel distinguere tinta, saturazione, e intensità del colore di una immagine. É risaputo, infatti, che l'HVS è molto meno sensibile a cambiamenti di tinta e saturazione che a cambiamenti di intensità. Tenendo in considerazione di questo fatto, molto spesso nelle codifiche delle immagini, lo spazio RGB viene mappato, attraverso una trasformazione lineare, nello spazio luminanza-crominanza. Molto spesso viene utilizzato lo spazio YCbCr (ITU-R BT.601). La trasformazione lineare che lega i due spazi e la seguente:

$$\begin{bmatrix} x_Y \\ x_{C_b} \\ x_{C_r} \end{bmatrix} \triangleq \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & 0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} x_R \\ x_G \\ x_B \end{bmatrix}$$

Un'altra rappresentazione molto utilizzata, è la rappresentazione YUV. Tale rappresentazione è molto simile allo spazio YCbCr, infatti la componente Y è la stessa, mentre le componenti U e V vengono calcolate con dei coefficienti diversi.

Un video digitale, invece, si realizza attraverso una sequenza di immagini digitali (a colori o meno). Ogni immagine della sequenza viene comunemente chiamata "frame"e fa riferimento ad uno specifico istante temporale. L'occhio umano, per le sue caratteristiche, riesce a distinguere circa 20 immagini al secondo, per questo il numero di frame al secondo (framerates o fps) di un video digitale è tipicamente un numero compreso tra 25 e 30.

Figura 2.1: Esempio di immagine stereo.

2.2 Immagini e Video Multiview

Un'*immagine multiview* consiste in un insieme di due o più immagini di una stessa scena o soggetto prese da differenti punti di vista. Se il numero di immagini è pari a due si parla di immagini stereo e in figura 2.1 è possibile osservarne un esempio. Tipicamente tutte le viste di un set di immagini multiview hanno la medesima risoluzione e fanno riferimento allo stessa scena. L'acquisizione di immagini di questo tipo avviene attraverso un insieme di fotocamere oppure con una singola camera a cui vengono fatte delle roto-traslazioni in modo da avere differenti punti di vista. Nel secondo caso però la scena deve essere tempo-invariante, in modo da fotografare sempre la stessa scena in tutte le viste. Fattore molto importante diventa quindi la distribuzione delle telecamere e la loro posizione nello spazio. Tipicamente queste vengono disposte in modo equispaziato su una o due dimensioni; molto spesso infatti vengono disposte su un arco di circonferenza o a formare una forma sferica nel caso bi-dimensionale. In figura 2.2 è mostrato un esempio di un sistema multiview composto da un set di 125 telecamere. Come per le immagini, il video multiview altro non è che una sequenza di immagini multiview. Da un punto di vista operativo può essere visto come un set di N video sincronizzati, dove N rappresenta il numero delle telecamere.

Figura 2.2: Stanford Multiview Camera Setup.

2.3 Dalla Scena all'Immagine

La formazione di un'immagine avviene attraverso un apparato che raccogliendo la luce riflessa della scena crea una immagine bidimensionale. Per riuscire ad ottenere informazioni della scena dall'immagine, è indispensabile capire come funziona questo processo. Esistono vari modelli che permettono di descrive la formazione dell'immagine, quello più comunemente usato, soprattutto per la sua semplicità, è sicuramente il modello della telecamera *pinhole*[1].

2.3.1 Modello della Telecamera *Pinhole*

Questo modello si basa sul principio di collinearità della *camera oscura* rinascimentale ed è rappresentato in figura 2.3 . Ogni punto **M** tridimensionale della scena, viene proiettato nel piano immagine R (o retina) nel punto **m** attraverso la retta che passa per il punto stesso e il foro stenopeico C (chiamato anche *centro ottico* o *centro di proiezione*). La distanza f tra il piano immagine e il centro C viene detta *distanza focale* mentre il *piano focale* F è quel piano parallelo a R contenente C. La linee ortogonale al piano immagine passante per C è detta *asse ottico* (in figura 2.3 coincide con l'asse

[1] Letteralmente: fotocamera a foro di spillo, viene comunemente tradotto in letteratura come foro stenopeico.

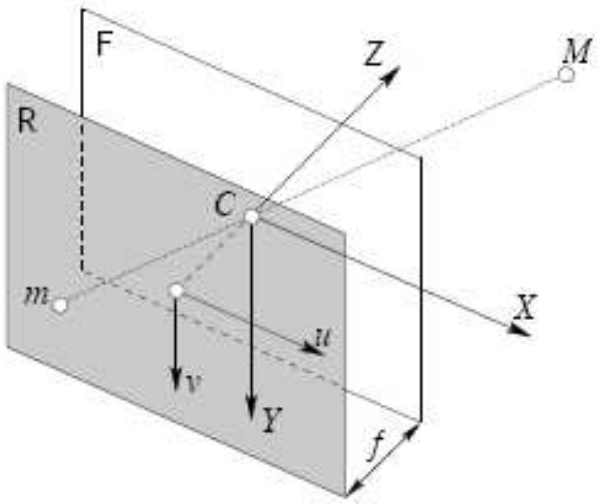


Figura 2.3: Modello della telecamera pinhole.

Z) e il suo punto p^* di intersezione con il piano immagine prende il nome di *punto principale*. Dalla similitudine dei triangoli di ottiene:

$$\frac{f}{z} = -\frac{u}{x} \quad \text{e} \quad \frac{f}{z} = -\frac{v}{y} \tag{2.1}$$

e quindi

$$\begin{cases} u &= \frac{-f}{z}x \\ v &= \frac{-f}{z}y \end{cases} \tag{2.2}$$

Per descrivere in modo compatto e pratico queste relazioni, risulta conveniente fare uso delle coordinate omogenee [4]. Definendo $\mathbf{m}$ e $\mathbf{M}$ come

$$\mathbf{m} \triangleq \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \quad \mathbf{M} \triangleq \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2.3}$$

si ha che la trasformazione dalle coordinate 3D a quelle 2D diventa lineare:

$$z\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} -fx \\ -fy \\ z \end{bmatrix} = \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2.4}$$

che in notazione matriciale assume la seguente forma

$$z\mathbf{m} = \mathcal{P}\mathbf{M} \tag{2.5}$$

oppure

$$\mathbf{m} \equiv \mathcal{P}\mathbf{M} \tag{2.6}$$

infatti

$$\begin{bmatrix} u \\ v \\ z \end{bmatrix} \equiv \begin{bmatrix} u/z \\ v/z \\ 1 \end{bmatrix} \tag{2.7}$$

ovvero i due vettori sono equivalenti, o meglio sono "uguali a meno di un fattore di scala "[4]. La matrice $\mathcal{P}$ rappresenta il modello geometrico della telecamera e viene chiamata *matrice della telecamera* o *matrice di proiezione prospettica* (MMP).

In un contesto più generale e realistico esistono alcuni fattori che portano a leggere modifiche al modello sopra descritto; infatti generalmente si ha:

- $C \neq (0, 0, 0)$
- L'asse ottico non coincidente con l'asse **Z**
- $p^* \neq (0, 0)$
- "Pixelizzazione"
- Effetti di distorsioni

I primi due punti dipendo principalmente da dove di trova la telecamera rispetto al sistema di rifermento e gli "aggiustamenti" che tengono conto di queste problematiche vengono definiti come *parametri estrinseci*. Gli altri punti invece dipendono da come è fatta fisicamente la telecamera, per questo vengono introdotti dei parametri detti *parametri intrinseci*.

2.3.2 Parametri Intrinseci

Considerando p_v e p_u le dimensioni dei pixel del sensore del dispositivo di acquisizione, W e H le sue dimensioni e il punto principale p^*, espresso in pixel, pari a $p^* = (W/2, H/2) = (u_0, v_0)$ la relarione (2.2) si modifica in questo modo:

$$\begin{cases} u &= \left(\frac{1}{p_u}\right)\left(\frac{-f}{z}\right)x + u_0 \\ v &= \left(\frac{1}{p_v}\right)\left(\frac{-f}{z}\right)y + v_0 \end{cases} \tag{2.8}$$

Definendo

$$k_u \triangleq \left(\frac{1}{p_u}\right) \quad \text{e} \quad k_v \triangleq \left(\frac{1}{p_v}\right)$$

la matrice di proiezione prospettica $\mathcal{P}$ diventa

$$\mathcal{P} = \begin{bmatrix} -fk_u & 0 & u_0 & 0 \\ 0 & -fk_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \mathcal{K}[\mathcal{I}|\mathbf{0}] \tag{2.9}$$

dove

$$\mathcal{K} = \begin{bmatrix} -fk_u & 0 & u_0 \\ 0 & -fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.10}$$

rappresenta la matrice dei parametri intrinseci.

Considerando inoltre che gli assi u e v non sono mai perfettamente ortogonali, viene introdotto un ulteriore parametro ovvero l'angolo θ tra questi due assi. La matrice $\mathcal{K}$ verrà così modificata

$$\mathcal{K} = \begin{bmatrix} -fk_u & fk_u\text{cotg}\theta & u_0 \\ 0 & -fk_v\text{sin}\theta & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.11}$$

2.3.3 Parametri Estrinseci

I parametri estrinseci introducono la trasformazione rigida che lega il sistema di riferimento del mondo esterno con quello standard della telecamera. Se si considera un punto nel sistema di riferimento mondo questo avrà delle coordinate omogenee pari a $\mathbf{M}$. Lo stesso punto nel sistema di riferimento

standard della telecamera avrà delle altre coordinate $\mathbf{M}_c$. La relazione che lega i valori delle coordinate tra i due sistemi sarà una roto-traslazione e si può scrivere nel seguente modo:

$$\mathbf{M}_c = G\mathbf{M} \tag{2.12}$$

dove

$$G = \begin{bmatrix} \mathcal{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \tag{2.13}$$

Essendo, per la (2.9)

$$\mathbf{m} = \mathcal{K}[\mathcal{I}|\mathbf{0}]\mathbf{M}_c \tag{2.14}$$

sostituendo la (2.12) si ha:

$$\mathbf{m} = \mathcal{K}[\mathcal{I}|\mathbf{0}]G\mathbf{M} \tag{2.15}$$

e dunque

$$\mathcal{P} = \mathcal{K}[\mathcal{I}|\mathbf{0}]G = \mathcal{K}[\mathcal{R}|\mathbf{t}] \tag{2.16}$$

2.4 Dall'Immagine alla Scena

Per passare dall'immagine alla scena, oltre a tutti i parametri della camera è necessaria un'ulteriore e importante informazione: la profondità. Cioè deve essere nota la distanza d tra il punto $\mathbf{Q}$ nello spazio e la sua proiezione $\mathbf{q}$ nell'immagine. La relazione che lega $\mathbf{Q}$ e $\mathbf{q}$ sarà quindi pari a [4]:

$$\mathbf{Q} = \mathcal{R}^{-1}(\mathcal{K}^{-1}\mathbf{q}d - \mathbf{t}) \tag{2.17}$$

dove la distanza d tra il piano dell'immagine e la scena viene solitamente fornita attraverso delle mappe di profondità (*depth map*).

2.4.1 Depth Map

La mappa di profondità di un'immagine è una matrice bidimensionale, solitamente della stessa dimensioni e risoluzione dell'immagine a cui fa riferimento. Ogni suo valore è proporzionale alla distanza tra il centro ottico e

il relativo punto della scena. I valori di una mappa di profondità, non rappresentano quindi direttamente la distanza, ma sono dei valori quantizzati e mappati su 8 o 16 bit tramite opportune funzioni di mappatura. In accordo con le convenzioni si ha che al punto più vicino verrà assegnato il valore 2^{B-1} mentre a quello più lontano il valore 0. Per avere la distanza è quindi necessario fare un'operazione di mappatura inversa. Non essendoci nessuna convenzione sulle funzioni di mappatura vengono qui riportate le due funzioni inverse utilizzate durante i test:

$$d \triangleq \frac{1}{\frac{z}{2^B-1}\left(\frac{1}{z_{near}} - \frac{1}{z_{far}}\right) + \frac{1}{z_{far}}} \tag{2.18}$$

$$d \triangleq z_{far} - \frac{z}{2^B - 1}\left(z_{far} - z_{near}\right) \tag{2.19}$$

dove B è il numero di bit per Pixel, z è il valore nella mappa e z_{near} e z_{far} rappresentano rispettivamente la distanza minima e massima dei punti rappresentabili dello spazio, dalla telecamera.

La figura 2.4 mostra un'immagine con la sua mappa di profondità a 8-bit.

(a) Immagine (b) Depth Map

Figura 2.4: Un'immagine a colori e la relativa depth map. Immagini del dataset "breakdancers" [3].

In un video o immagine multiview si parla di "multi-view plus depth" dove vengono memorizzate le informazioni di profondità vista per vista e frame per frame.

2.5 Valutazione della codifica

La valutazione della bontà di una codifica video può essere fatta attraverso degli indici oggettivi oppure attraverso degli indici soggettivi. Per poter fare dei confronti tra codifiche solitamente vengono usate delle misure oggettive.

MSE

L'errore quadratico medio (MSE, *Mean-Squared Error*) tra l'immagine originale x e l'immagine decodificata $\hat{x}$ viene definito come:

$$\text{MSE} = \frac{1}{n}\sum_{x=0}^{n}(x_i - \hat{x}_i)^2 \tag{2.20}$$

dove n indica il numero totale di campioni considerati.

PSNR

Il *peak-signal-to-noise-ratio* (PSNR) è dato da

$$\text{PSNR} = 10\log_{10}\left(\frac{V_{MAX}^2}{\text{MSE}}\right) \tag{2.21}$$

dove MSE è l'errore quadratico medio (2.20) mentre V_{MAX} è il massimo valore che può assumere un pixel all'interno dell'immagine. Se si considerano immagini a 8 bit $V_{MAX} = 2^8 - 1 = 255$.

Questa misura è espressa in decibel (dB) e tipicamente assume valori in un range compreso tra 28 e 42 dB. Il PSNR va solitamente calcolato in funzione del bit rate, in questo modo è possibile fare dei confronti tra diversi algoritmi di compressione, valutandone così l'efficienza. Più alto è il PSNR e migliore sarà la qualità dei dati compressi. Questo indice, in quanto un misura oggettiva, è sicuramente utile per fare dei confronti tra algoritmi differenti, ma non da nessuna informazione, per quanto riguarda la qualità delle immagini, dal punto di vista della percezione umana. É infatti risaputo, ad esempio, che alcuni tipi di distorsioni che comportano un abbassamento dei

valori di PSNR non danno nessun effetto alla percezione umana e viceversa. Per risolvere questo tipo di problema, in letteratura vengono proposte misure che tengono in considerazione la qualità come percezione umana, uno di questi è il SSIM *structural similarity index* introdotto da Wang *et al.* in [2].

In questa tesi comunque la misura di qualità a cui si fa riferimento è il PSNR, comunemente usato in letteratura.

Capitolo 3

Fondamenti di Teoria delle Wavelet

In questo capito verrà fatta una breve introduzione teorica sulla teria delle Wavelet. Nella seconda parte verrà invece mostrato come queste trasformate possono essere applicate nella compressione delle immagini.

3.1 La Trasformata Wavelet

Le trasformate wavelet sono classificate a livello generale nella trasformata wavelet continua (*Continuous Wavelet Transform*, CWT) e nella trasformata wavelet discreta (*Discrete Wavelet Transform*, DWT). La differenza di principio fra le due è il fatto che la trasformata continua opera su tutte le possibili scale e traslazioni, mentre la trasformata discreta usa un sottoinsieme discreto di tutti i valori possibili.

Per semplicità di esposizione e fornire un inquadramento generale, vengono ora esposte alcune definizioni e nozioni fondamenti delle wavelet riferite a funzioni di una sola varabile. L'estensione a funzioni di più variabile è comunque abbastanza elementare e si attua come per la trasformata di Fourier.

3.1.1 La trasformata wavelet continua

Si considera un funzione continua $f(t)$ a energia finita

$$\int |f(t)|^2 dt < \infty. \tag{3.1}$$

ovvero $f \in \mathbf{L}^2(R)$.

Si definisce *trasformata wavelet continua* di $f(t)$ (*continous wavelet transform*, CWT) la funzione

$$\gamma(s,\tau) = \int f(t)\psi^*_{s,\tau}(t)dt \tag{3.2}$$

dove

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\psi\left(\frac{t-\tau}{s}\right). \tag{3.3}$$

L'equazione (3.2) mostra come una funzione $f(t)$ venga decomposta in un insieme di funzioni elementari $\psi_{s,\tau}(t)$, le quali prendono il nome di *wavelets*. La trasformata wavelet, inoltre, è una funzione a due variabili, s e t, che rappresentano rispettivamente *scala* e *traslazione*. Dall'equazione (3.3) si può inoltre osservare come tutte le wavelets vengano generate da una singola funzione principale $\psi(t)$, detta *wavelet madre*, attraverso uno scalamento e una traslazione, mentre il fattore $1/\sqrt{s}$ è un fattore di normalizzazione dell'energia della wavelet.

Per completezza si definisce inoltre la *trasformata inversa*, che viene definita come segue

$$f(t) = \int\int \gamma(s,\tau)\psi_{s,\tau}(t)d\tau ds. \tag{3.4}$$

Principali proprietà

Le principali proprietà delle wavelets sono l'*ammisibilità* e le *condizioni di regolarità*.

Detta $\Psi(\omega)$ la trasformata di Fourier della wavelet madre, essa deve soddifare la *condizione di ammissibilità* [6]

$$\int \frac{|\Psi(\omega)|^2}{|\omega|} d\omega < +\infty. \tag{3.5}$$

Tale condizione implica che la trasformata di fourier $\Psi(\omega)$ sia nulla alla frequenza zero

$$|\Psi(\omega)|^2\Big|_{\omega=0} = 0 \tag{3.6}$$

Ciò significa che le wavelet devono avere un adamenento di tipo passa-banda. Inoltre avere valore nullo alla frequenza zero comporta che nel dominio del tempo il valore medio delle wavelet deve essere nullo

$$\int \psi(t) dt = 0. \tag{3.7}$$

In altri termini le wavelet devono essere funzioni oscillanti.

Le condizioni di regolarita invece sono concetti abbastanza complessi, legati alla differenziabilità delle wavelet. Sono condizioni supplementari che vengono imposte alle funzioni wavelet, al fine di rendere i coefficienti della trasformata legati alla variabile s, in modo da farli diminuire rapidamente con la diminuzione della scala.

In linea di princio il grado di regolarità q di una wavelet è dato dal suo ordine massimo di derivabilità. Per avere una regolarità maggiore di n, una wavelet deve avere almeno $(n+1)$ momenti nulli (v. Daubechies, 1988). Un aspetto particolarmente importante della regolarità sono quindi i *"vanishing moments"*, ovvero la proprietà della wavelet madre di avere i primi n momenti nulli:

$$M_p = 0 \quad \text{per} \quad p = 0, 1, \ldots, n. \tag{3.8}$$

Infatti se si considera lo sviluppo in serie di Taylor in $t = 0$ della traformata wavelet (3.2) fino all'ordine n si ha (per semplicità $\tau = 0$)

$$\gamma(s, 0) = \frac{1}{\sqrt{s}} \left[\sum_{p=0}^{n} f^{(p)}(0) \int \frac{t^p}{p!} \psi\left(\frac{t}{s}\right) dt + O(n+1) \right] \tag{3.9}$$

dove con $f^{(p)}$ si intende la derivate p-esima di f, e considerando la definizione di momento

$$M_p = \int t^p \psi(t) dt \tag{3.10}$$

la (3.9) diventa

$$\gamma(s,0) = \frac{1}{\sqrt{s}} \left[\sum_{p=0}^{n} \frac{f^{(p)}(0)}{p!} M_p s^{p+1} + O(s^{n+2}) \right] \tag{3.11}$$

se i primi n momenti sono a zero ciò significa che i coefficienti $\gamma(s,\tau)$ decadranno a zero come s^{n+2}. Va notato inoltre che, grazie alla condizione di ammissibilità, ogni wavelet possiede almento un vanishing moments, ovvero il momento di primo ordine (3.7).

3.1.2 Wavelets Discrete

La trasformata wavelet continua viene calcolata su spostamenti e scalature continue del segnale e tipicamente il numero di wavelet è infinito. Questo fatto porta la trasformazione CWT ad essere poco adatta ad una implementatazione al calcolatore ma soprattutto risulta essere molto ridondante.

Le *wavelets discrete* vengono quindi definite considerando solo scalature e traslazioni con passi discreti. L'equazione (3.3) viene quindi modificata nella seguente forma

$$\psi_{j,k}(t) = \frac{1}{\sqrt{s_0^j}} \psi \left(\frac{t - k\tau_0 s_0^j}{s_0^j} \right) \tag{3.12}$$

dove j e k sono numeri interi.
Il parametro s_0 viene definito come *passo di scala* ed è un valore prefissato maggiore di 1 ($s_0 > 1$); tipicamente si sceglie $s_0 = 2$ in modo che il campionamento dell'asse delle frequenze corrisponda a un campionamento *diadica* [7]. Il parametro τ_0, invece è il *passo di traslazione* che dipende dal passo di scala e solitamente si seglie pari a 1.L'effetto della discretizzazione delle wavelet è perciò quello di campinare lo spazio "tempo-scala" a intervalli discreti come in fugura. 3.1 .

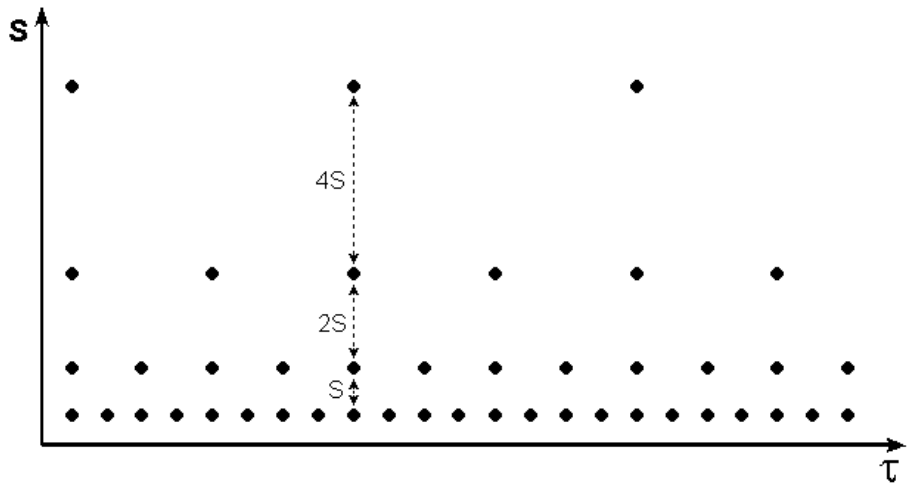


Figura 3.1: Localizzazione delle wavelets discrete nello spazio "tempo-scala" su una griglia *diadica* [7].

L'utilizzo delle wavelet discrete nella CWT porterà come risultato una serie di coefficienti wavelet che prendono il nome di *wavelet series decomposition*. Per poter ricostruire il segnale da questa decomposizione, condizione necessaria e sufficiente è che l'energia dei coefficienti wavelet deve essere compresa tra due limiti positivi [8]

$$A\|f\|^2 \leq \sum_{j,k} |\langle f, \psi_{j,k}\rangle|^2 \leq B\|f\|^2 \tag{3.13}$$

dove $\|f\|^2$ è l'energia di $f(t)$, e $A > 0$, $B < \infty$ indipendenti da f.
Se la (3.13) è soddisfatta la famiglia di wavelet discrete $\psi_{j,k}(t)$, con j e k interi, prende il nome di *frame* con limite A e B. Se A e B coicidono, il frame viene detto *tight*, e le wavelet diventano una base ortonormale del segnale f. Dal momento che le wavelet sono discrete, possono essere rese ortogonali scegliendo una wavelet madre tale per cui

$$\int \psi_{j,k}(t)\psi^*_{m,n}(t) = \begin{cases} 1 & \text{se} \quad j = m \quad \text{e} \quad k = n \\ 0 & \text{altrove} \end{cases} \tag{3.14}$$

In questo modo un qualsiasi segnale può essere ricostruito dalla somma pesata

dei coefficienti della CWT [6]

$$f(t) = \sum_{j,k} \gamma(j,k)\psi_{j,k}(t) \tag{3.15}$$

L'ortogonalità non è comunque essenziale nella rappresentazione del segnale. In alcune applicazioni, infatti, il fatto di non essere ortogonali, e la conseguente ridondanza, possono aiutare a ridurre la sensibilità al rumore [6] o a migliorare lo "shift invariance" della trasformata [9].

Le wavelet discrete hanno comunque bisogno di un numero infinito di scalature e traslazioni per poter essere utilizzate per calcolare la trasformata wavelet. La via più semplice, per avere la trasformata in una forma utilizzabile, è sicuramente quella di prendere un numero finito di wavelet discrete. Il problema che si presenta ora, però, è capire quante wavelet bisogna considerare.

Data la natura del segnale da analizzare (3.1), di sicuro esiste un limite nelle traslazioni delle wavelet. Per quanto riguarda le scale,invece, si considera il fatto che le wavelet hanno uno spettro di tipo passa-banda e perciò dalla teoria delle trasformate di Fourier una compressione nel tempo corrisponde a una espansione in frequenza:

$$\mathcal{F}\{f(at)\} = \frac{1}{|a|} F\left(\frac{w}{a}\right) \tag{3.16}$$

In questo modo possiamo ricoprire lo spettro del segnare utilizzando solo le wavelet che ci interessano, come avviene per le traslazioni nel dominio del tempo. In pratica è come avere un banco di filtri. D'altro canto però, la natura di tipo passa-banda delle wavelet, non permette di comprire tutta la banda del segnale. Infatti se ad esempio si considera un fattore di stretch pari a due nel tempo la banda delle wavevlet viene di volta in volta dimezzata, coprendo sempre metà dello spettro rimasto. Ciò significa, in teoria, che comunque servono un numero di wavelet infinito.

3.1.3 Scaling function

La *scaling function* $\varphi(t)$ viene introdotta da Mallat [10] e consente di "raggirare" il problema della copertura dello spettro alle basse frequenze. La scaling function non è altro che un filtro passa-basso che fa da tappo (cork) e va a coprire quelle frequenze che non vengono raggiunte dalle wavelet a causa della loro natura di tipo passa-banda come mostrato in figura 3.2 .

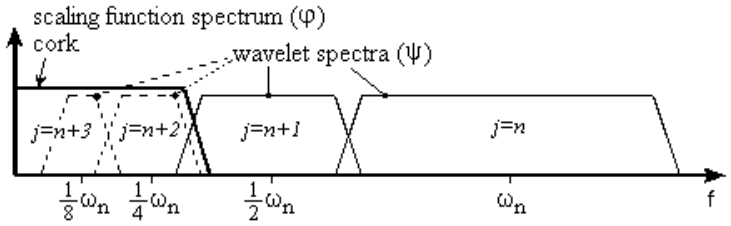


Figura 3.2: La Scaling function fa da "tappo".

La scaling function è comunque una funzione e può essere quindi espressa attraverso la decomposizione wavelet (3.15):

$$\varphi(t) = \sum_{j,k} \gamma(j,k)\psi_{j,k}(t) \tag{3.17}$$

Tale decomposizione utilizza un numero infinito di wavelet ma a partire da una certa scala j, per via della natura passa-basso della funzione $\varphi(t)$. Questo significa che se viene fatto un uso combinato delle wavelet e della scaling function, ovvero fino a una certa scala j si usano le wavelet e poi tutte le altre vengono coperte dalla funzione $\varphi(t)$, si riesce ad avere la copertura totale dello spettro utilizzando un numero finito di wavelet più la scaling function.

3.1.4 Wavelet e Subband coding

Come già accennato nel paragrafo 3.1.2, le wavelet discrete possono essere viste come un banco di filtri. Se si considera il rapporto tra la frequenza centrale e la larghezza di banda di ogni wavelet si potrà osservare come tale

rapporto sia sempre lo stesso per tutte le wavelet. Questo rapporto se riferito a un filtro viene chiamato "*fidelity factor Q*" del filtro e perciò nel caso delle wavelet si parlerà di banco di filtri *constant-Q*

Il fatto di analizzare un segnale attraverso un banco di filtri è cosa assai nota, soprattutto nel mondo dalla visione computazionale, e prende il nome di *subband coding*. Solitamente si divide lo spettro del segnale in due parti uguali: bassa frequenza e alta frequanza. L'alta frequenza contiene pochi dettagli di interesse e perciò viene lasciata integra. La parte in bassa frequenza, soprattutto nelle immagini, contiene molta informazione, e viene nuovamante divisa in due parti uguali procendendo in modo iterativo come mostrato in figura 3.3 fino a quando è possibile (soprattutto da un punto di

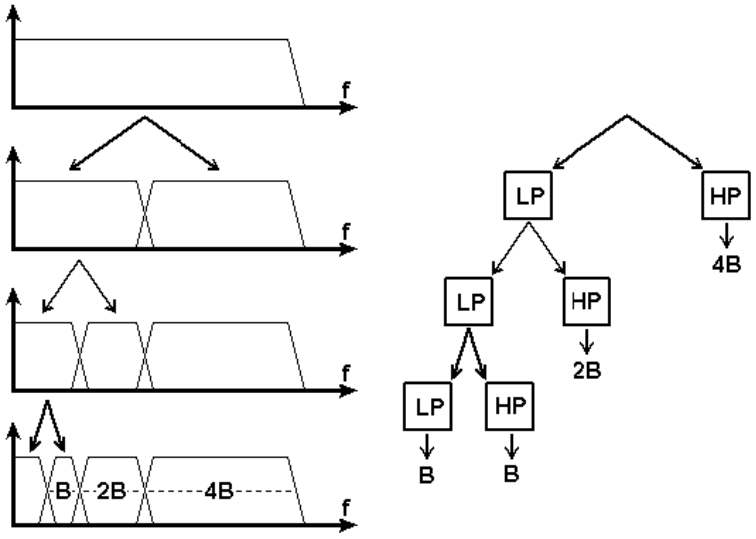


Figura 3.3: Split dello spettro del segnale con un banco di filtri iterativo.

vista computazionale).
Questo modo di procedere prende il nome di banco di filtri iterativo (*iterated filter bank*).

Osservando la figura 3.3 si nota come la subband coding sia equivalente a un filtro passa-basso e a un banco di filtri passa-banda, dove ogni filtro del banco ha una larghezza di banda raddoppiata rispetto al vicino di sinistra.

Per quanto detto all'inizio del paragrafo questo equivale a una trasformata wavelet del segnale e il filtro passa-basso reppresenta la scaling function. Si può quindi concludere che una trasformata wavelet essendo equivalente a un banco di filtri *constant-Q* può essere tranquillamente rappresentata attraverso uno schema di tipo subband coding [10].

3.1.5 La Trasformata Wavelet Discreta

In molti casi pratici, come ad esempio le immagini, il segnale di interesse risulta essere campionato. Le wavelets discrete sono funzioni discrete sia nella scala s e nella traslazione τ ma non nel tempo t.

La trasformata wavelet, come già affermato più volte, corrisponde a un banco di filtri. Una *traformata wavelet discreta* (DWT, discrete wavelet transform) corrisponderà quindi, in modo intuitivo, ad un *banco di filtri digitali*. La dimostrazione di tale affermazione non è di facile comprensione; per una trattazione completa sull'argomento si rimanda a [7].

3.2 Compressione di Immagini con Trasformata wavelet

Uno dei campi di maggior utilizzo delle trasformate wavelet è sicuramete la compressione di immagini digitali. Lo standard JPEG2000, il quale è stato progettato per aggiornare e sostituire lo standard JPEG, fa uso di trasformata wavelet al posto della DCT rendendo più efficiente la compressione. Nell'introdurre la trasformata wavelet si è fatto riferimento a segnali unidimensionali, le immagini però sono segnali bidimensionali. Esistono sostanzialmete due approcci per la decomposizione in sottobande delle immagini, e più in generale dei segnali a due dimensioni: il primo si basa su filtri bi-dimensionali, mentre il secondo usa trasformate separabili che possono essere quindi implementate con filtri a una dimensione, prima lungo le

righe e poi lungo le colonne o viceversa. La maggior parte degli algoritmi di compressione, tra cui anche il JPEG2000, solitamente fanno uso del secondo approcio.

In figura 3.4 viene mostrato come un'immagine può essere decomposta

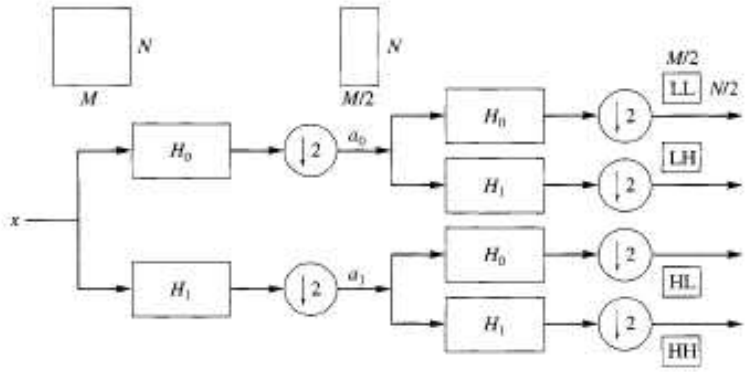


Figura 3.4: Decomposizione in sottobande di un'immagine $N \times M$

in sottobande. Partendo da un'immagine $N \times M$ ogni riga viene filtrata e sottocampinata ottenendo due immagini $N \times \frac{M}{2}$, succesivamente vengono filtrate e sottocampionate le colonne ottenendo quattro immagini $\frac{N}{2} \times \frac{M}{2}$. Le quattro immagini ottenute corrispondono alle possibili combinazioni di filtraggio delle righe e delle colonne e vengono definite nel seguente modo:

- **LL image -** passa *basso* lungo le righe e passa *basso* lungo le colonne
- **LH image -** passa *basso* lungo le righe e passa *alto* lungo le colonne
- **HL image -** passa *alto* lungo le righe e passa *basso* lungo le colonne
- **HH image -** passa *alto* lungo le righe e passa *alto* lungo le colonne

Questa decomposizione viene molto spesso rappresentata in un'unica immagine come motrato in figura 3.5 . Ognuna delle sotto-immagini ottenute da questo modello, può essere filtrata e sottocampionata in modo da ottenerne

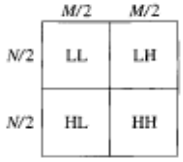
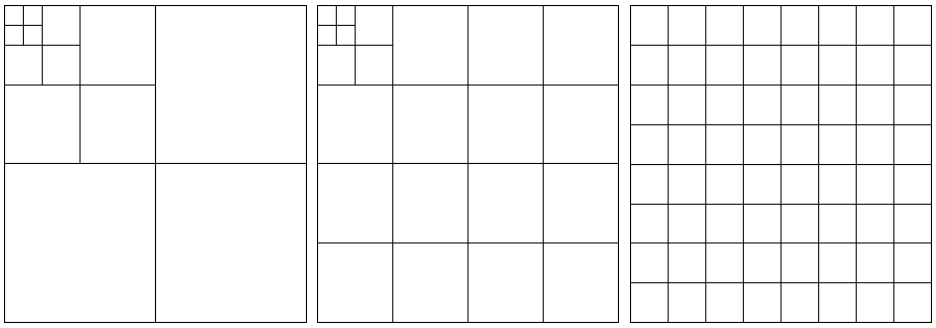


Figura 3.5: Rappresentazione del primo livello di decomosizione

altre quattro. Questo processo può quindi continuare fino ad ottenere la sottobanda desiderata. Si possono quindi realizzare varie strutture in funzione delle esigenze e alcune di queste sono mostrate in figura 3.6. La figura 3.7 invece mostra i primi tre livelli di decomposizione per l'immagine Lena [1] attraverso lo schema di decomposizione mostrato in figura 3.6(a): la sottobanda LL viene decomposta, dopo ogni decomposizione, in quattro sotto-immagini ottenendo un totale di 10 sottobande.



(a) (b) (c)

Figura 3.6: Tre esempi di strutture di decomposizione in sottobande

Grazie alla tecnica delle wavelet è possibile ottenere delle immagini compresse scalabili sia in risoluzione che in distorsione. Lo schema di figura 3.6(a) infatti permette la scalabilità in risoluzione dato che è possibile rappresentare le sottobande di ogni livello, senza alcuna dipendenza nei confronti dei livelli di risoluzione più alti. Soprattutto grazie a questa proprietà di scalabilità lo schema di figura 3.6(a) è quello scelto per il JPEG2000.

[1] Lena è una delle immagini più comunemente usate per valutare gli algoritmi di compressione

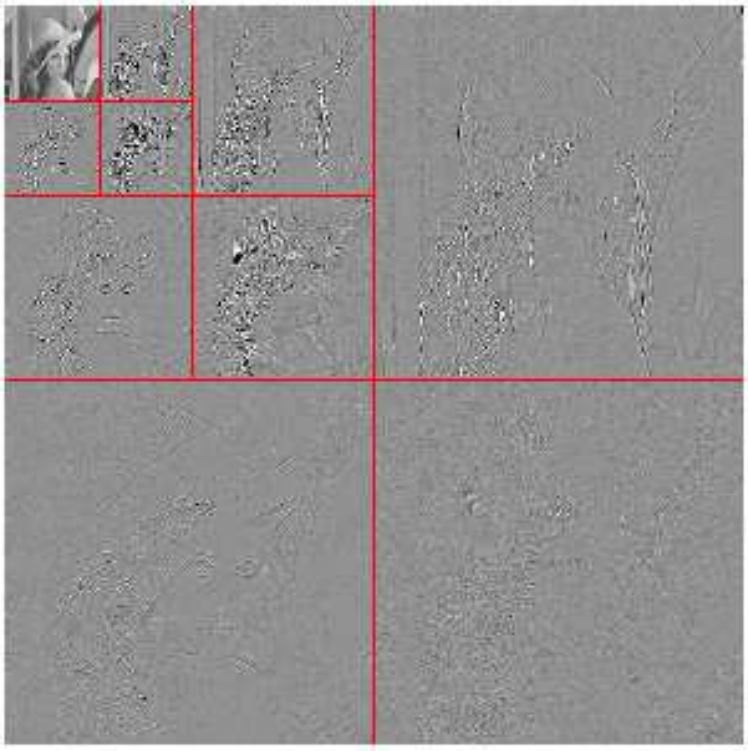


Figura 3.7: Esempio di tre livelli di decomposizione

3.2.1 JPEG2000

In questa sezione viene fatto una breve introduzione allo standard JPEG2000, il quale fa uso di trasformata wavelet.

Lo standard JPEG garantisce un eccellente rendimento per rate superiori allo 0.25 bit per pixel. Tuttavia a bassi bit rate la qualità dell'immagine è molto scadente. Per risolvere questo problema si è cominciato a lavorare su un altro standard, conoscito come JPEG2000, basato sulla decomposizione wavelet al posto della DCT (Discrete Cosine Transform). In figura 3.8 si può osservare come lo standard JPEG2000, a bassi bit rate, sia qualitativamente migliore dello standard JPEG a parità di bit rate.

Lo parte di codifica del JPEG2000 si basa sullo schema originariamente proposto da Taubman [19] e Taubman e Zakhor [20], conoscito come EBCOT

(a) Originale 703kB (b) JPEG 5kB (c) JPEG2000 5kB

Figura 3.8: La stessa immagine compattata con JPEG e JPEG2000 con lo stesso bit-rate.

(Embedded Block Coding with Optimized Truncation).

Le fasi fondamentali della codifica JPEG2000 possono essere schematizzate in

- Trasformazione delle componenti
- Trasformata wavelet
- Quantizzazione
- Codifica Entropica EBCOT

La figura 3.9 mostra le fasi dello standard in uno schema a blocchi mentre in

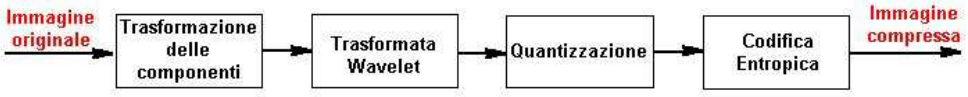


Figura 3.9: Shema a blocchi dello standard JPEG2000

figura 3.10 viene mostrata una rappresentazione grafica.

Nella prima fase l'immagine originale viene per prima cosa separata nei vari canali di colore. Nel formato JPEG2000 sono state definite due codifiche di colore di base, una reversibile (RCT - *Reversible Color Transform*) per la compressione loseless e l'altra irreversibile (ICT - *Irreversible Color Transform*) di tipo lossy e perfettamente analoga alla conversione RGB to YCbCr usata nel JPEG. Le immagini così ottenute possono essere trattate

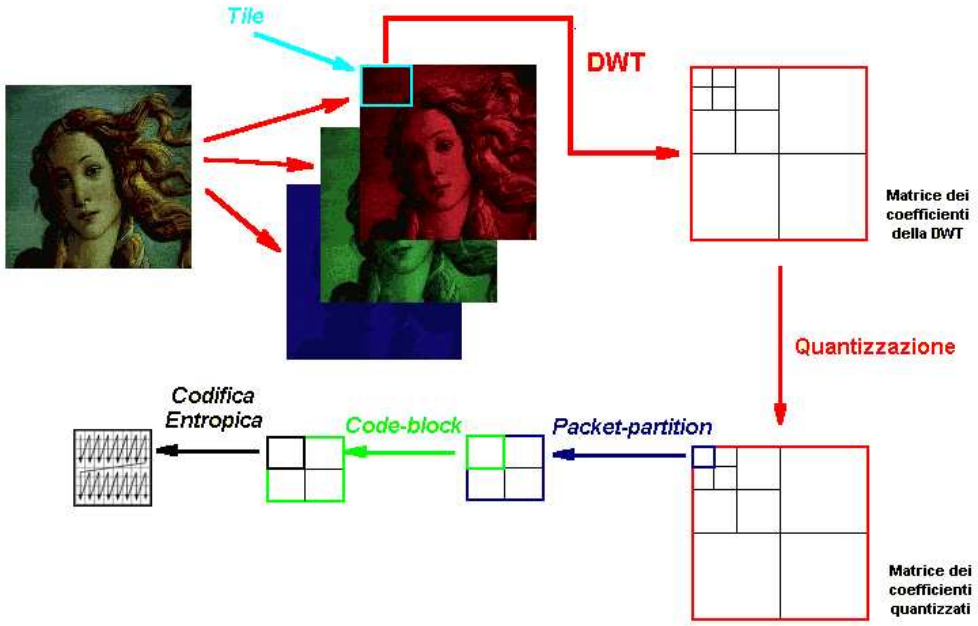


Figura 3.10: Rappresentazione grafica dello standard JPEG2000

indipendentemente come se fossero in livelli di grigio.
Prima di passare alla traformata waveletet i vari liveli di colore vengo scomposti in *tile* [2], non sovrapposti, da trattare in modo indipendentemente.
Ad ogni tite viene quindi applicata separatamente la trasformata wavelet. La suddivisione dell'immagine in tile permette l'accesso diretto a una determinata regione dell'immagine (utile, ad esempio, per uno zoom), ed essendo possibile elaborare separatamente le tile, viene sensibilmente ridotta la memoria necessaria per la compressione. Dopo la trasformata i coefficienti wavelet ottenuti vengono quantizzati e raggruppati in *code-blocks*, che vengono codificati in maniera indipendente gli uni dagli altri con una codifica di tipo entropica.

Ciascun tile viene quindi trattato separatamente dagli altri e i parametri di compressione possono essere diversi da tile a tile. Questo viene definito come codifica ROI (Region Of Interest), ovvero la possibilità di salvare con

[2] Tile: letteralmente piastrelle

risoluzione diverse dal resto dell'immagine quelle zone ritenute più importanti.

Capitolo 4

Algoritmo 1
3D Warping e Wavelet Lifting

In questo capitolo viene descritto un nuovo approccio per applicare le wavelet nella codifica di immagini multiview. Il metodo utilizzato è quello introdotto da Zamarin *et al.* in [1] dove la 3D-DCT viene sostituita con la trasformata wavelet tipo Haar o 5/3.

4.1 Quadro Generale

In un insieme di immagini multiview la distribuzione delle videocamere gioca un ruolo molto importante per la realizzazione di una codifica efficiente; la correlazione tra le varie viste, infatti, dipende fortemente dal posizionamento delle videocamere. Risultano quindi di particolare interesse i parametri intrinseci ed estrinseci delle camere e soprattutto le informazioni geometriche della scena. Tali informazioni possono essere fornite attraverso una completa rappresentazione tridimensionale della scena oppure attraverso la depth map di ogni vista, come discusso nella sezione 2.4.1. In questa tesi si è fatto uso delle depth map in quanto sono comunemente usate nelle applicazioni 3D video, ma nulla vieta di usare un modello tridimensionale della scena.
I dati in input dell'algoritmo proposto saranno quindi:

- Immagini Multiview.
- Depth map di ogni vista.
- Parametri intrinseci ed estrinseci di ogni videocamera.

Un esempio di "multi-view plus depth" è fornito dalla sequenza "breakdancer" di Microsoft Research [3], comunemente usata per verificare le prestazioni degli algoritmo di compressione 3D. Questa sequenza contiene tutte le informazioni necessarie per l'algoritmo ed è composta da 8 stream video acquisiti da un insieme di videocamere posizionate in modo regolare come mostrato in figura 4.1 (a). Ad ogni immagine inoltre è associata una depth map come

(a) Scenario

(b) Prima immagine insieme Breakdance

Figura 4.1: Sequenza Breakdance

mostrato in figura 4.1 (b).

Lo schema di codifica proposto in [1] si basa fondamentalmente su due operazioni ovvero il 3D-Warping delle viste disponibili e la successiva applicazione della codifica 3D-DCT. La figura 4.2 mostra i passi principali dell'algoritmo.

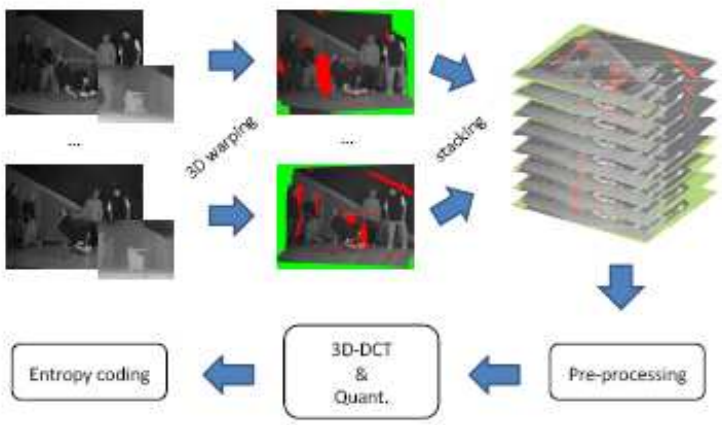


Figura 4.2: Step principali dell'algoritmo con 3D-DCT.

In questa tesi la seconda operazione, che fa uso della codifica 3D-DCT, viene sostituita della trasformata wavelet di tipo Haar o 5/3. In figura 4.3 viene riportato il nuovo schema analizzato.

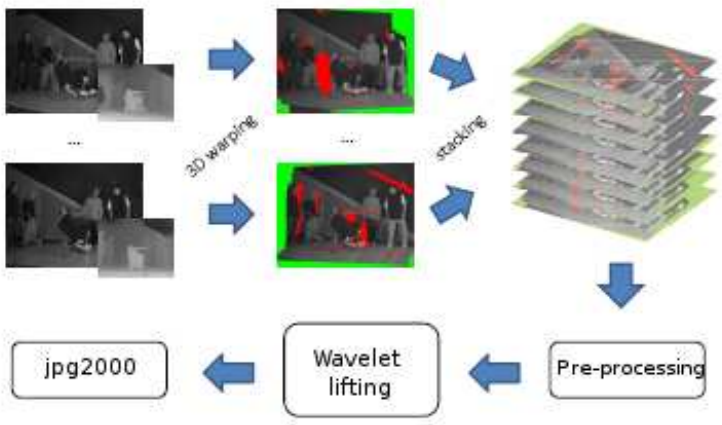


Figura 4.3: Step principali dell'algoritmo con l'utilizzo di Wavelet lifting.

Definedo con $V_i, i = 0, \ldots, k-1$, l'insieme delle immagini multiview da codificare, dove k è il numero di videocamere, l'algoritmo di codifica si divide in tre step principali:

1. **3D Warping -** In questo primo step tutte le viste vengono proiettate su una stessa vista di riferimento V_{ref} scelta tra le k viste disponibili.

In questo modo per ogni immagine V_i si ottiene una nuova immagine $V_{i \to ref}$ la quale rappresenta una "predizione" della vista di riferimento dalla vista i-esima. Nel fare questa operazione è possibile notare come in queste nuove immagini $V_{i \to ref}$ esistano delle regioni che non possono essere determinate. Tali regioni sono zone di occlusione o regioni "out-of-bound" (ad esempio regioni appartenenti al campo visivo di $V_{i \to ref}$ ma non di V_i) e verranno quindi trattate in modo separato. Il risultato di questo primo step è perciò un insieme di viste $V_{i \to ref}, i = 0, \ldots, k-1$ le quali vengo memorizzate in un unica struttura che chiameremo "*image-stack*" e che contiene la maggior parte dei dati da codificare.

2. **Wavelet Lifting -** In questo secondo step viene applicato uno schema di wavelet lifting tra le k viste $V_{i \to ref}, i = 0, \ldots, k-1$ ottenendo $k-1$ matrici $H_i, i = 0, \ldots, k-2$ rappresentanti i coefficienti in alta frequenza e una matrice L_0 con i coefficienti in bassa frequenza. Tali matrici vengono poi compresse con bit rate differenti in quanto la matrice in bassa frequenza contiene più informazioni rispetto a quelle in alta che possono quindi essere compresse a bit rate più bassi.

3. **Codifica delle occlusioni -** In questo ultimo step vengono gestite le zone di occlusione ovvero quelle regioni che non sono visibili nella vista di riferimento v_{ref}.

4.2 Architettura di Codifica

Come già anticipato nel paragrafo precedente l'algoritmo si divide in tre step principali:

1. 3D Warping.

2. Wavelet lifting.

3. Codifica delle occlusioni.

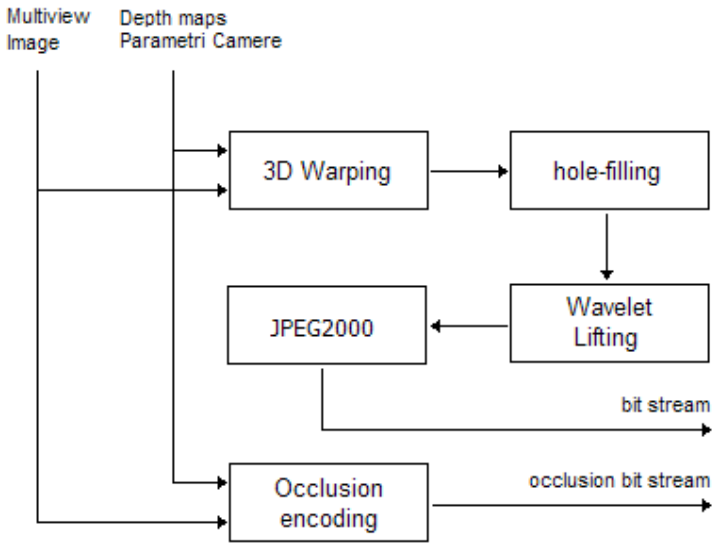


Figura 4.4: Diagramma a blocchi dell'algoritmo proposto.

La figura 4.4 mostra un diagramma a blocchi dell'algoritmo proposto. Ogni blocco verrà ora descritto in modo dettagliato.

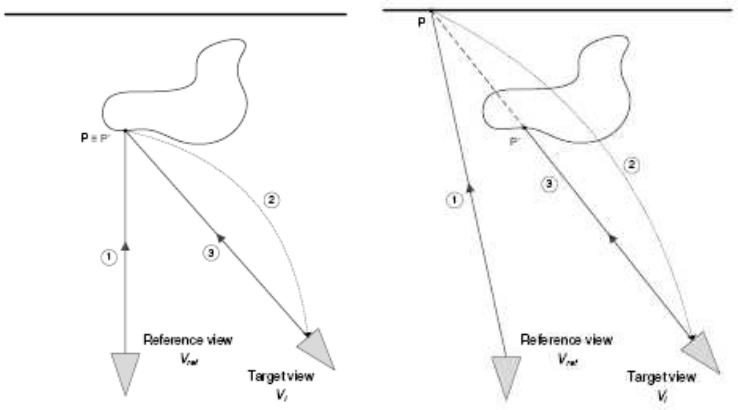
4.2.1 3D Warping

Questa fase consiste nel fare il "3D warping" di ogni vista V_i nella vista di riferimento V_{ref} utilizzando le informazioni di profondità e i parametri delle camere. Attraverso queste informazioni è possibile infatti calcolare il punto tridimensionali $\mathbf{P}$, nel "sistema di riferimento mondo", corrispondente a ogni pixel $\mathbf{p_i}$ di ogni immagine.

Il 3D-warping può essere fatto essenzialmente in due modi. Nel primo caso ogni pixel di ogni vista V_i viene mappato nella vista di riferimento V_{ref} attraverso l'uso delle depth map D_i delle viste (*forward mapping*). Nel secondo caso invece i pixel dalla vista di riferimento V_{ref} vengono mappati, attraverso D_{ref}, nelle varie viste V_i (*backward mapping*). Le regioni di occlusione che si creano nei due approcci sono diverse e perciò devono essere trattate in modo diverso per non perdere informazioni. In questo lavoro, per la creazione dello stack di immagini, si è fatto uso essenzialmente della tecnica backward map-

ping (che utilizza sostanzialmente le informazioni della mappa di profondità di riferimento D_{ref}) mentre per la rivelazione delle occlusioni si è fatto uso anche dell'approccio forward mapping.

Di seguito viene descritta la procedura di warping usata per la creazione dell'*image-stack* (figura 4.5).



(a) (b)

Figura 4.5: Esempio di warping corretto (a) e di occlusione (b). Nel caso (a) $\mathbf{P}$ coincide con $\mathbf{P'}$ perciò il punto è correttamente visto in entrambe le immagini. Nel caso (b) invece i due punti non coincidono; si è in presenza di una occlusione e perciò il pixel della vista target V_i non può essere usato per il warping.

Per prima cosa indichiamo con Π_{ref} la matrice di proiezione corrispondente alla vista V_{ref} e per semplicità di notazione definiamo

$$d_{ref} \triangleq D_{ref}(\mathbf{p_{ref}})$$

il valore di profondità associato al pixel $\mathbf{p_{ref}}$.
Le fasi dell'operazione di warping possono essere così suddivise:

1. Applicando l'equazione (2.17) è possibile, per ogni pixel $\mathbf{p_{ref}}$, trovare il corrispondente punto tridimensionale $\mathbf{P}$ nello spazio.

2. Attraverso il modello della camera pinhole (paragrafo 2.3) il punto $\mathbf{P}$ viene mappato nel punto $\mathbf{p_i}$ corrispondente nella vista V_i. Come discusso nel paragrafo 2.3.1, se Π_i rappresenta la matrice si proiezione corrispondente alla vista V_i, $\mathbf{p_i}$ sarà semplicemente pari a $\mathbf{p_i} = \Pi_i \mathbf{P}$.

3. Essendo disponibili le mappe di profondità per ogni vista è possibile verificare in modo semplice le occlusioni. Calcolando infatti il punto $\mathbf{P'}$ associato a $\mathbf{p_i}$ attraverso la depth map D_i di V_i è possibile valutare la presenza di occlusioni semplicemente analizzando la distanza tra i due punti $\mathbf{P}$ e $\mathbf{P'}$. Se non ci sono occlusioni $\mathbf{P}$ e $\mathbf{P'}$ sono lo stesso punto, come mostrato in figura 4.5(a), e la loro distanza è quindi nulla. Nel caso in cui i due punti non coincidano allora significa che si è in presenza di occlusione. Un esempio è mostrato chiaramente in figura 4.5(b). Essendo le mappe di profondità non perfette ma affette da piccoli errori e imperfezioni di acquisizione, anche se dal punto di vista teorico $\mathbf{P}$ e $\mathbf{P'}$ dovrebbero coincidere, nel caso non siano presenti occlusioni, i due punti possono risultare vicini ma non coincidenti. Per ovviare a questo tipo di inconveniente si è introdotta una soglia per poter valutare se i due punti rappresentano lo stesso punto. Se la distanza tra i due punti è inferiore a questa soglia si assumono i due punti coincidenti.

In figura 4.6 (a) è riportata $V_{0\to4}$ ovvero la prima immagine della se-

(a) $V_{0\to4}$ (b) V_4

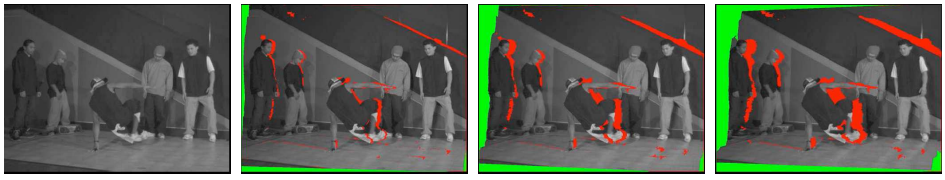
Figura 4.6: Proiezione della prima immagine rispetto alla vista centrale.

quenza "breakdancers" V_0 ottenuta dopo il warping rispetto all'immagine V_4 (figura 4.6(b)). I pixel rossi rappresentano le zone occluse mentre quelle verdi rappresentano le zone "out-of-bound".

Una volta che tutte le immagine $V_{i\to ref}$ vengono calcolate queste vanno a formare una matrice tridimensionale che rappresenta l'image-stack. Un

esempio delle immagini che compongono l'image-stack è mostrato in figura 4.7 sempre relativo alla sequenza breakdancers.

(a) $V_{0\to4}$ (b) $V_{1\to4}$ (c) $V_{2\to4}$ (d) $V_{3\to4}$



(e) $V_{4\to4}$ (f) $V_{5\to4}$ (g) $V_{6\to4}$ (h) $V_{7\to4}$

Figura 4.7: Tutte le viste che formano l'image-stack per la sequenza "breakdancers".

Per rendere la codifica più efficiente in [1] prima di passare alla codifica viene applicato un processo di riempimento delle zone occluse e out-of-bound. Tale processo di riempimento risulta molto utile e rende la codifica migliore anche per questo nuovo schema con wavelet.

Linear Filling Process.

La soluzione proposta in [1] si basa su un' interpolazione lineare tra i pixel delle viste. Se si considerano 8 viste, come per la sequenza "breakdancers", un vettore di 8 pixel è definito per ogni posizione (x, y). I pixel a zero verranno riempiti con una interpolazione lineare dei pixel non a zero che li circondano. In figura 4.8 è mostrato un esempio di riempimento lineare dove in rosso sono evidenziati i pixel calcolati in quanto a zero perché riferiti a zone occluse o out-of-bound. La figura 4.9 mostra lo stack delle immagini dopo il processo di riempimento delle occlusioni. Da notare è il fatto che almeno un pixel è sempre diverso da zero, che è quello della vista di riferimento.

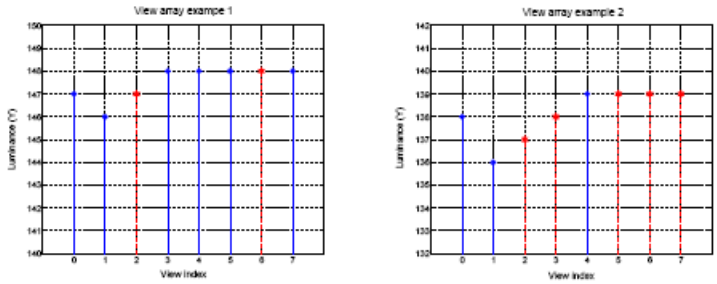


Figura 4.8: Esempio di riempimento lineare delle zone occluse.

(a) $V^f_{0\to4}$ (b) $V^f_{1\to4}$ (c) $V^f_{2\to4}$ (d) $V^f_{3\to4}$

(e) $V^f_{4\to4}$ (f) $V^f_{5\to4}$ (g) $V^f_{6\to4}$ (h) $V^f_{7\to4}$

Figura 4.9: L'image-stack "breakdancers" dopo il processo di riempimento.

4.2.2 Wavelet Lifting

In generale la struttura lifting genera una decomposizione in sottobande in alta (H, High-pass) e bassa (L, Low-Pass) frequenza, attraverso una sequenza di passi di "predizioni" (P, prediction) e "aggiornamenti" (U, update) che rappresentano la stima e compensazione della disparità tra le immagini. In letteratura si possono trovare molte strutture di wavelet lifting, in questa tesi sono state analizzate la Haar e la 5/3.

Alle k viste che formano l'image-stack viene quindi applicato uno schema di tipo wavelet lifting Haar o 5/3. Lo schema introdotto in questa fase è essenzialmente lo stesso che verrà analizzato nel prossimo capitolo relativo

alla *Disparity-Compensated Wavelet Lifting* (paragrafo 5.2.1). In questo caso però le immagini da elaborare non sono direttamente le immagini multiview ma le k viste che formano l'image-stack. In questo schema di lifting perciò le operazioni di predizione e aggiornamento che costituiscono la DCWL (paragrafo 5.2.1) e che consentono di compensare la "disparità" che sussiste tra le viste adiacenti, vengono meno in quanto sono già state pre-elaborate nella prima fase relativa al 3D-Warping. Ogni vista dell'image-stack infatti rappresenta l'immagine di riferimento.

Se si considera la versione Haar l 'i-esima componente in bassa frequenza (L_i) e l'i-esima in alta frequenza (H_i), possono essere scritte nel seguente modo

$$H_i = V^f_{2i+1\to ref} - a_{2i,2i+1} V^f_{2i\to ref} \tag{4.1}$$

$$L_i = V^f_{2i\to ref} + b_{2i+1,2i} H_i \tag{4.2}$$

mentre per la versione 5/3

$$H_i = V^f_{2i+1\to ref} - a_{2i,2i+1} V^f_{2i\to ref} - a_{2i+2,2i+1} V^f_{2i+2\to ref} \tag{4.3}$$

$$L_i = V^f_{2i\to ref} + b_{2i-1,2i} H_{i-1} + b_{2i+1,2i} H_i \tag{4.4}$$

Dove $V^f_{i\to ref}$ sono le immagini in uscita dal processo di filling. Come si può osservare la operazioni di prediction e update non vengono fatte perché intrinseche nelle immagini. I fattori di scala $a_{m,n}$ e $b_{m,n}$ usati rispettivamente in P e U assumono i valori indicati nella tabella 4.1. I valori assunti dai fattori di scala sono stati calcolati considerando lo schema descritto in [16]. Nello specifico, $a_{m,n}$ è pari all'inverso del numero di viste che entrano in gioco nel calcolo della wavelet mentre $b_{m,n} = a_{m,n}/2$.

Tipo	$a_{m,n}$	$b_{m,n}$
Haar	1	$\frac{1}{2}$
5/3	$\frac{1}{2}$	$\frac{1}{4}$

Tabella 4.1: Fattori di scala nei due tipi di wavelet lifting.

Se si considera un insieme di k immagini multiview il massimo numero di livelli possibili di wavelet lifting sarà pari $\log_2 k$. Per ottenere quindi un

insieme di coefficienti altamente efficienti dal punto di vista dello step successivo (ovvero la compressione) è bene sfruttare tutti i livelli possibili dello schema. Se si prende in esame l'insieme di riferimento "breakdancers", che contiene 8 immagini, il massimo numero di livelli sarà pari a 3 come mostrato in figura 4.10 .

In figura 4.11 viene riportato l'insieme delle immagini che verranno codificate attraverso l'algoritmo JPEG2000[1], il quale fa uso delle trasformate wavelet. Come si nota chiaramente nella figura, tutta l'informazione, o quasi, dopo il processo di lifting viene "trasferita" nell'immagine a bassa frequenza mentre le immagini in alta frequenza non contengono molta informazione e perciò possono essere compattate a bit-rate più bassi.

Embeded Filling Process.

Per rendere ancora più efficiente il processo di riempimento delle zone occluse, e aumentare leggermente il rendimento dell'algoritmo, è stato implementato un processo di riempimento ad hoc, alternativo al linear filling proccess, strettamente legato alla struttura di lifting delle wavelets. La presenza di zone occluse o out-of-bound nelle immagini rappresentanti l'image-stack, senza il processo di linear filling, renderebbe la wavelet lifting poco efficiente in quanto le zone occluse si propano nei coefficienti ad alta frequenza, rendendo di fatto quasi inutili gli effetti della wavelet. In figura 4.11, si può notare come le immagini in alta frequenza, senza il processo di filling, contengano molta più informazione rispetto a quelle della figura 4.11.

La procedura realizzata semplicemente recupera i dati mancanti, dovuti alle zone di occlusione o out-of-bound del 3dWarping, prendendoli dalle viste che vengono prese in esame per il calcolo della wavelet. Se si considera la wavelet di tipo Haar, i dati delle zone occluse della vista $V_{2i \to ref}$ verranno recuperati, dove possibile, dalla vista $V_{2i+1 \to ref}$ e viceversa, aggiungendo un peso pari a $a_{m,n}$. Per quanto riguarda la wavelet 5/3 questi dati vengono

[1]Si è fatto uso del codificatore KAKADU.

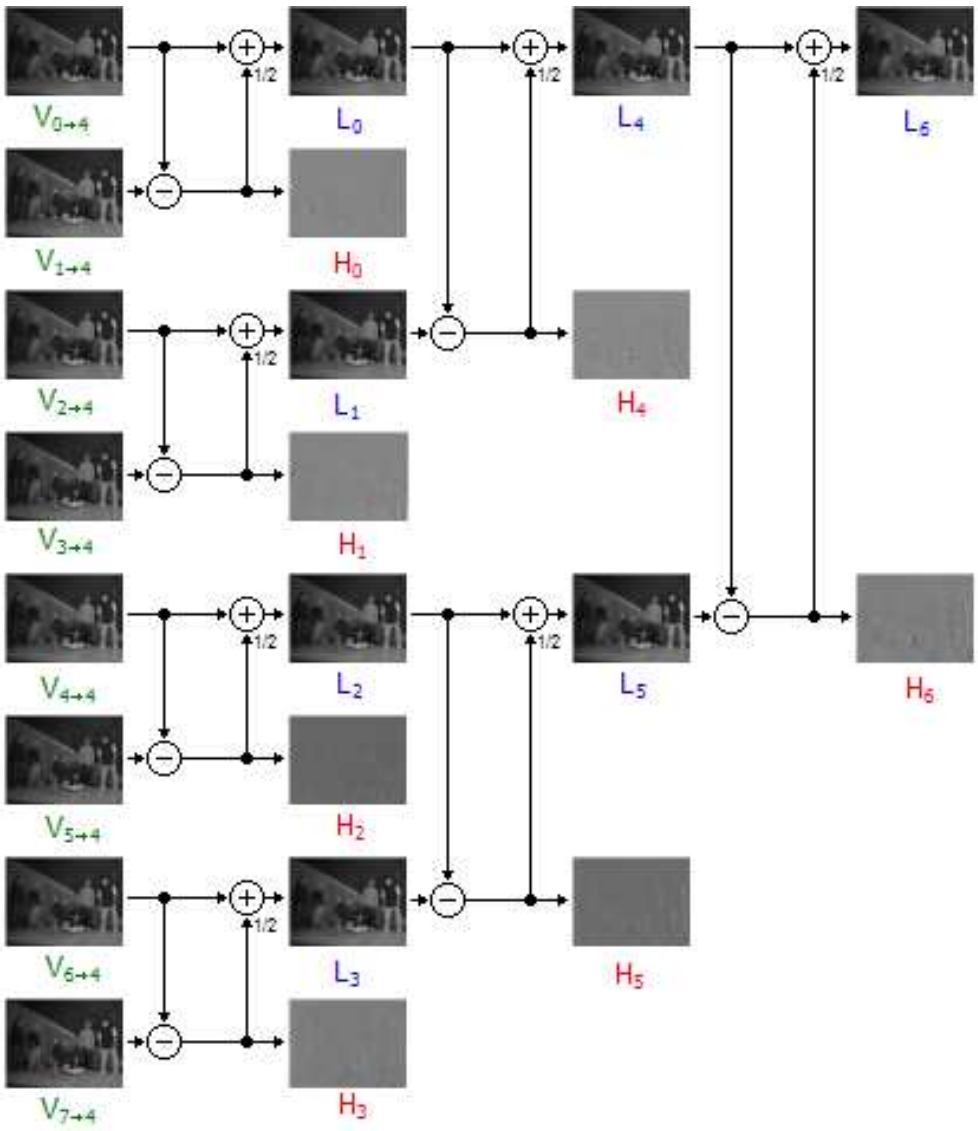
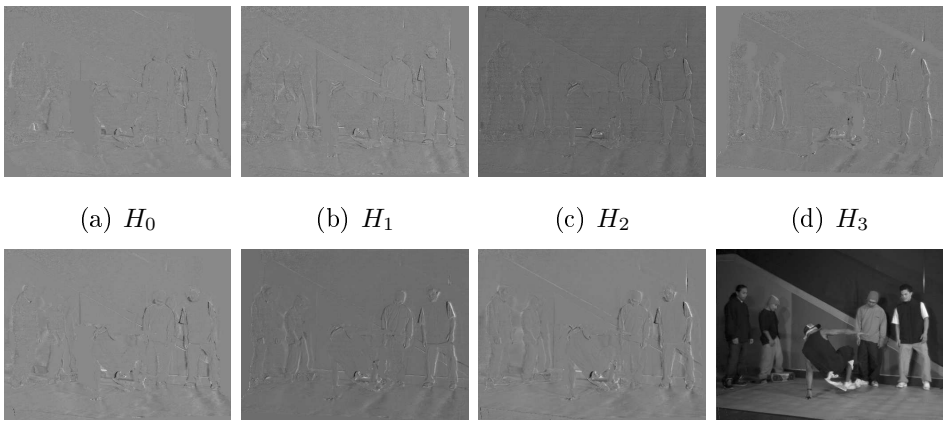


Figura 4.10: Tutti i livelli del Wavelet Lifting tipo haar dell'image-stack relativo alla sequenza "breakdancers"



(a) H_0 (b) H_1 (c) H_2 (d) H_3

(e) H_4 (f) H_5 (g) H_6 (h) L_6

Figura 4.11: L'image-stack "breakdancers" dopo il wavelet lifting tipo Haar.

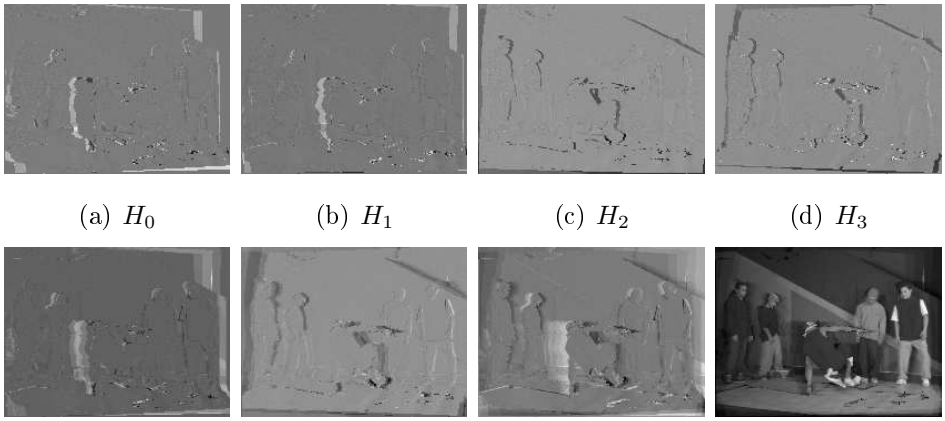
recuperati come somma delle due viste adiacenti $V_{2i+1 \to ref}$ e $V_{2i-1 \to ref}$ sempre con peso $a_{m,n}$.In figura 4.13 viene mostrato come cambia lo schema lifting per la wavelet tipo Haar con l'utilizzo di questa procedura.

Il procedimento appena illustrato non migliora di molto l'efficienza dell'algoritmo in termini di PSNR o bit rate, anzi i risultati dimostrano che le curve praticamente sono coincidenti. Un miglioramento si ha invece in termini computazionali. Questo approccio infatti rende l'algoritmo più veloce in quanto si lavora localmente nelle zone di interesse e l'interpolazione avviene in modo indiretto con l'avanzamento dei livelli dello schema wavelet.

4.2.3 Codifica delle occlusioni

La gestione delle occlusione si basa sostanzialmente su due step. In una prima fase, le occlusioni e le out-of-bound di ogni vista, vengono raggruppate in un unica immagine formata da blocchetti di 8x8 pixel, come mostrato in figura 4.14. Dopo questa fase il secondo step consiste nel compattare in modo efficiente tale immagine.

L'immagine in figura 4.14 si ottiene partendo dalle immagini relative alle zone di occlusione e out-of-bound, ad esempio per la sequenza "breakdancers"



(a) H_0 (b) H_1 (c) H_2 (d) H_3

(e) H_4 (f) H_5 (g) H_6 (h) L_6

Figura 4.12: L'image-stack "breakdancers" dopo il wavelet lifting tipo Haar senza filling proccess.

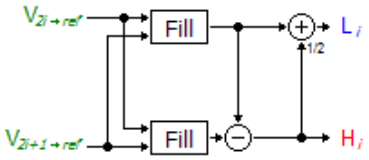


Figura 4.13: Nuovo schema con Embeded Filling Process per wavelet lifting tipo Haar.

tali immagini sono mostrate in figura 4.15 e vengono create considerando blocchetti di 8x8 pixel.

Per diminuire la quantità di informazione viene inoltre applicato un processo di warping per verificare se queste zone sono ricavabili dalle immagini precedenti o successive. Tale processo in [1] viene definito come Inter-view filling process. Nello specifico la prima immagine V_0 rimane inalterata. La seconda immagine, V_1, viene modificata togliendo tutti quei punti che possono essere ricavati dalla prima. Dalla terza immagine, invece, viene tolta tutta quella informazione che può essere ricavata dalla prima e dalla seconda immagine e così via per le altre immagini fino a V_{ref}. Per le immagini successive la procedura avviene in modo simmetrico partendo dalla vista più lontana da V_{ref}. In figura 4.16 viene illustrato lo schema dell'inter-view filling process relativo a 8 viste mentre l'immagine 4.17 mostra tutte le immagini

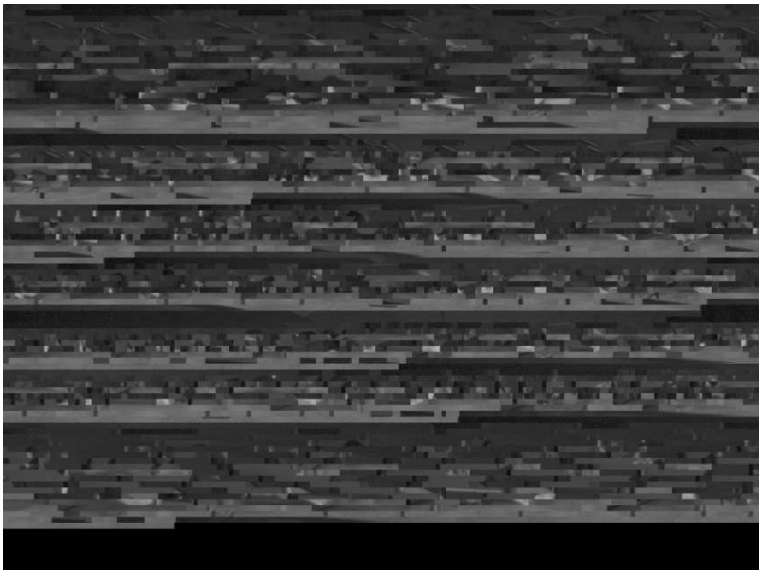
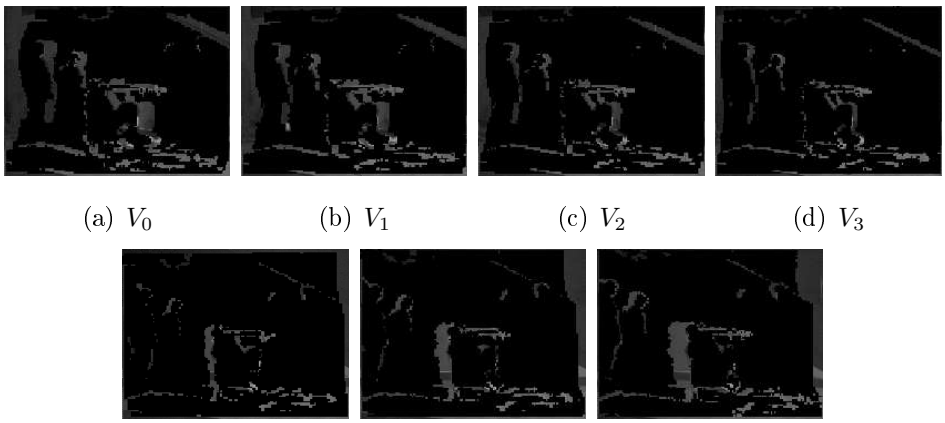


Figura 4.14: Immagine delle occlusioni in blocchi 8x8 pixel relativa alla sequenza "breakdancers".

della sequenza breakdancers dopo tale processo.

In figura 4.18 viene mostrato in particolare la seconda immagine della sequenza breakdancers prima e dopo l'inter-view filling process; in questa immagine si può osservare chiaramente come, dopo tale processo, l'informazione sia notevolmente diminuita garantendo perciò una compressione a rate più bassi a parità di qualità.

Per comprimere l'immagine finale 4.14 si è fatto uso del classico standard JPEG, infatti, dopo prove sperimentali, questa scelta è risultata migliore rispetto al JPEG2000, dovuta al fatto di avere blocchetti di 8x8 pixel. La codifica JPEG infatti lavorando su blocchetti di dimensione 8x8 non crea artefatti sui bordi tra un blocchetto e l'altro in quanto ogni blocchetto è compresso in modo indipendente cosa che non si verifica con JPEG2000.



(a) V_0 (b) V_1 (c) V_2 (d) V_3

(e) V_5 (f) V_6 (g) V_7

Figura 4.15: Insieme delle immagini delle zone di occlusione e out-of-bound della sequenza "breakdancers"

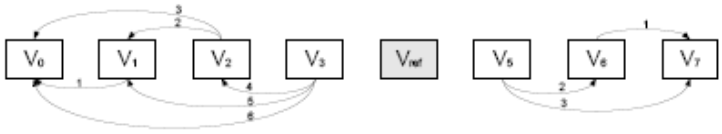
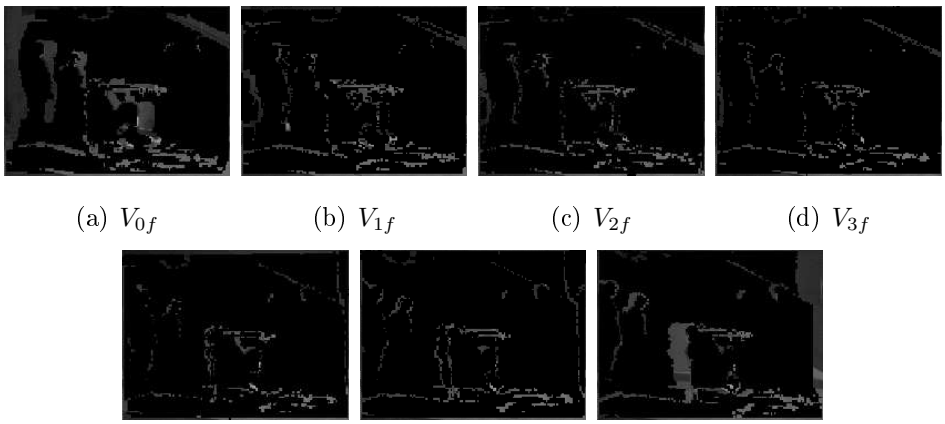


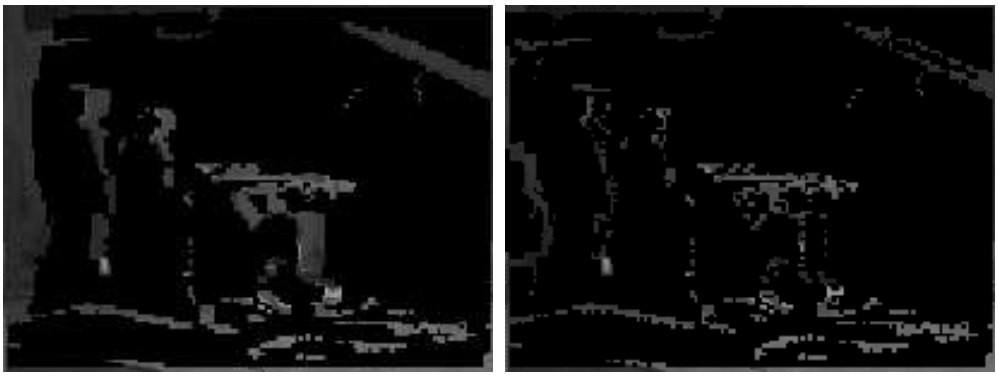
Figura 4.16: Schema dell'inter-view filling process relativo a 8 viste. Il numero sulle frecce sta a indicare l'ordine con il quale avviene il processo di filling



(a) V_{0f} (b) V_{1f} (c) V_{2f} (d) V_{3f}

(e) V_{5f} (f) V_{6f} (g) V_{7f}

Figura 4.17: Insieme delle immagini delle zone di occlusione e out-of-bound della sequenza "breakdancers" dopo l'inter-view filling process



(a) V_1 (b) V_{1f}

Figura 4.18: Seconda immagine delle occlusioni per la sequenza "breakdancers" (a) prima e (b) dopo l'inter-view filling process

Capitolo 5

Algoritmo 2 Disparity-Compensated Wavelet Lifting

L'obbiettivo principale degli algoritmi di compressione di immagini multiview, che si possono trovare in letteratura, è sostanzilamente quello di eliminare il più possibile la ridondanza tra le varie viste in modo da ridurre al minimo l'infomazione necessaria per ricostrure le immagini. In questo contesto la trasformata wavelet viene utilizzata attraverso tecniche che vengono dette "*tecniche di wavelet lifting*"[1] e vengono applicate principalmente tra le viste adiacenti.

5.1 Quadro Generale

Come per l'algoritmo descritto nel capitolo 4, anche qui la distribuzione delle videocamere gioca un ruolo molto importante per una codifica efficiente. I parametri intrinseci ed estrinseci delle camere e supprattutto le informazioni geometriche della scena risultano di particolare interesse. I dati in input del-

[1] Tecniche proposte e applicate per la prima volta nel multiview in [11, 12].

l'algoritmo proposto saranno quindi gli stessi dell'altro algoritmo e vengono di seguito elencati:

- Immagini Multi-view.
- Depth map di ogni vista.
- Parametri intrinseci ed estrinseci di ogni videocamera.

Lo schema generale dell'algorimo risulta molto semplice ed è mostrato in figura 5.1). L'insieme delle immagini multiview vengono codificate attre-

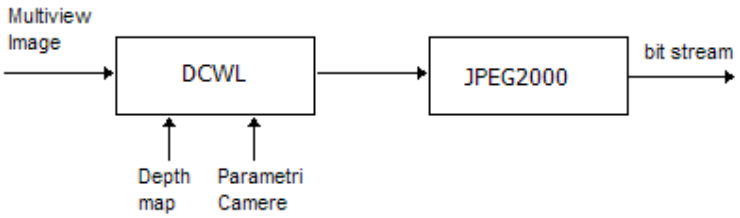


Figura 5.1: Step principali dell'algoritmo.

verso una struttura di wavelet lifting chiamata *Disparty-Compensated Lifting*(DCWL), la quale al suo interno fa delle operazioni di warping tra le viste adiacenti, e successivamente i coefficienti ottenuti vengono compressi con un codificatore JPEG2000[2].

5.2 Wavelet Lifting

La struttura wavelet lifting realizza la trasformata wavelet con una struttura molto simile alla subband coding (paragrafo 3.1.4). Nella codifica video classica la tecnica di wavelet lifting viene efficientemente usata per la compensazione del moto [14] [15] e per questo viene comunemente definita come *Motion-Compendated Temporal Filtering*(MCTF) . Nella compressione di immagini multiview, invece, tale tecnica viene applicata tra le viste adiacenti

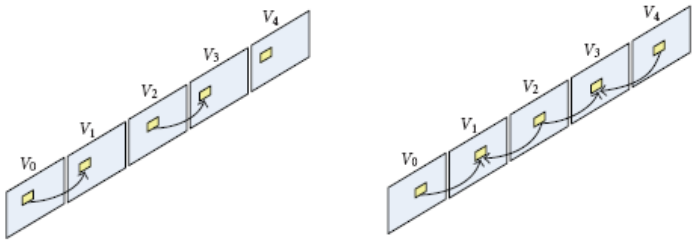
[2]Si è fatto uso del codificatore KAKADU

in modo da compensare la "disparità" che sussiste tra queste, e perciò viene comunemente chiamata *Disparty-Compensated Lifting*(DCWL).

5.2.1 Disparity-Compensated Wavelet Lifting

Codifica

La codifica wavelet lifting (*analysis side*) decompone le immagini MultiView nelle sottobande H e L. Supponiamo di avere una sequenza di immagini multi-view formate da N viste. Tali immagini possono essere divise nelle viste pari X_{2i}, e nelle viste dispari X_{2i+1}, con $i = 0, \ldots, \lfloor N/2 \rfloor$, che in generale presentano un alto grado di correlazione. Nella DCWL di tipo Haar, la compensazione della disparità si realizza usando solo una delle due immagini adiacenti alla vista di riferimento, mentre per quanto riguarda la versione 5/3 entrambe le viste vengono utilizzate. Nello specifico la DCWL di tipo Haar userà la vista $i-1$ o la vista $i+1$ per ridurre la ridondanza nella vista i, mentra la DCWL 5/3 le userà entrambe.
La figura 5.2 mostra come vengono referenziate le immagini per la DCWL



(a) Haar (b) 5/3

Figura 5.2: Illustrazione delle viste di riferimento nella DCWL Haar e 5/3.

di tipo Haar e 5/3.

Per la DCWL di tipo Haar, l 'i-esima componente in bassa frequenza (L_i) e l'i-esima in alta frequenza (H_i), possono essere scritte come

$$H_i = V_{2i+1} - a_{2i,2i+1} P(V_{2i}, \hat{d}_{2i+1 \to 2i}) \tag{5.1}$$

Tipo	$a_{m,n}$	$b_{m,n}$
Haar	1	$\frac{1}{2}$
5/3	$\frac{1}{2}$	$\frac{1}{4}$

Tabella 5.1: Fattori di scala nei due tipi di wavelet lifting.

$$L_i = V_{2i} + b_{2i+1,2i} U(H_i, \hat{d}_{2i \to 2i+1}) \tag{5.2}$$

L 'i-esima componente in bassa frequenza (L_i) e l'i-esima in alta frequenza (H_i), per quanto riguarda la DCWL di tipo 5/3, la quale utilizza entrambe le viste adiacenti per eseguire la compesazione di disparità, possono essere scritte come

$$H_i = V_{2i+1} - a_{2i,2i+1} P(V_{2i}, \hat{d}_{2i+1 \to 2i}) - a_{2i+2,2i+1} P(V_{2i+2}, \hat{d}_{2i+1 \to 2i+2}) \tag{5.3}$$

$$L_i = V_{2i} + b_{2i-1,2i} U(H_{i-1}, \hat{d}_{2i \to 2i-1}) + b_{2i+1,2i} U(H_i, \hat{d}_{2i \to 2i+1}) \tag{5.4}$$

$P(V_m, \hat{d}_{n \to m})$ è lo step di predizione, con il quale viene calcolata la compensazione di disparità della immagine V_m su V_n e corrisponde al calcolo del warping dalla vista n alla vista m ovvero, usando la sintassi del capitolo precedente,

$$P(V_m, \hat{d}_{n \to m}) = V_{m \to n}$$

.

$U(H_k, \hat{d}_{l \to j})$ invece è lo step di aggiornamento e viene calcolato con H_k. Dal punto di vista implementativo i due step di prediction e update corrispondono quindi alle funzioni di warping tra le varie viste. In pratica nella fase di predizione si cerca di togliere l'informazione ricavabile dall'immagine adiacente, per quanto riguarda Haar, o dalle due immagini adiacenti per il 5/3,ottenedo così il segnale in alta-frequenza. Tale predizioni è perciò ottenibile proiettando (warping) le informazioni da una vista a quella adiacente. I fattori di scala $a_{m,n}$ e $b_{m,n}$ vengono calcolati con lo stesso criterio utilizzato nel paragrafo 4.2.2 ovvero lo schema definito in [16], e i valori assunti vengono riportati in tabella 5.1.

Decodifica

La parte di decodifica (*synthesis side*) ricostruiche le immagini multiview partendo dalle sottobande H e L ma invertendo gli step U e P della codifica. Le immagini ricostruite della DCWL di tipo Haar assumono la seguente forma

$$V_{2i}^{'} = L_i^{'} - b_{2i+1,2i} U(H_i^{'}, \hat{d}_{2i \to 2i+1}) \tag{5.5}$$

$$V_{2i+1}^{'} = H_i^{'} + a_{2i,2i+1} P(V_{2i}^{'}, \hat{d}_{2i+1 \to 2i}) \tag{5.6}$$

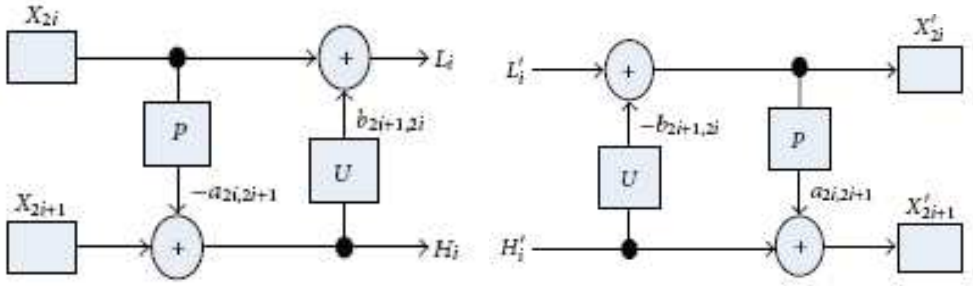
Per quanto riguarda la DCWL di tipo 5/3 si ha

$$V_{2i}^{'} = L_i^{'} - b_{2i-1,2i} U(H_{i-1}^{'}, \hat{d}_{2i \to 2i-1}) - b_{2i+1,2i} U(H_i^{'}, \hat{d}_{2i \to 2i+1}) \tag{5.7}$$

$$V_{2i+1}^{'} = H_i^{'} + a_{2i,2i+1} P(V_{2i}^{'}, \hat{d}_{2i+1 \to 2i}) + a_{2i+2,2i+1} P(V_{2i+2}^{'}, \hat{d}_{2i+1 \to 2i+2}) \tag{5.8}$$

dove $V_{2i}^{'}$ e $X_{2i+1}^{'}$ sono le versioni ricostruite della viste V_{2i} e V_{2i+1} le quali appunto possono essere leggermente diverse a causa delle perdite che ci possono essere dovute all'agoritmo di compressione utilizzato per comprimere L e H nel nostro caso JPEG2000.

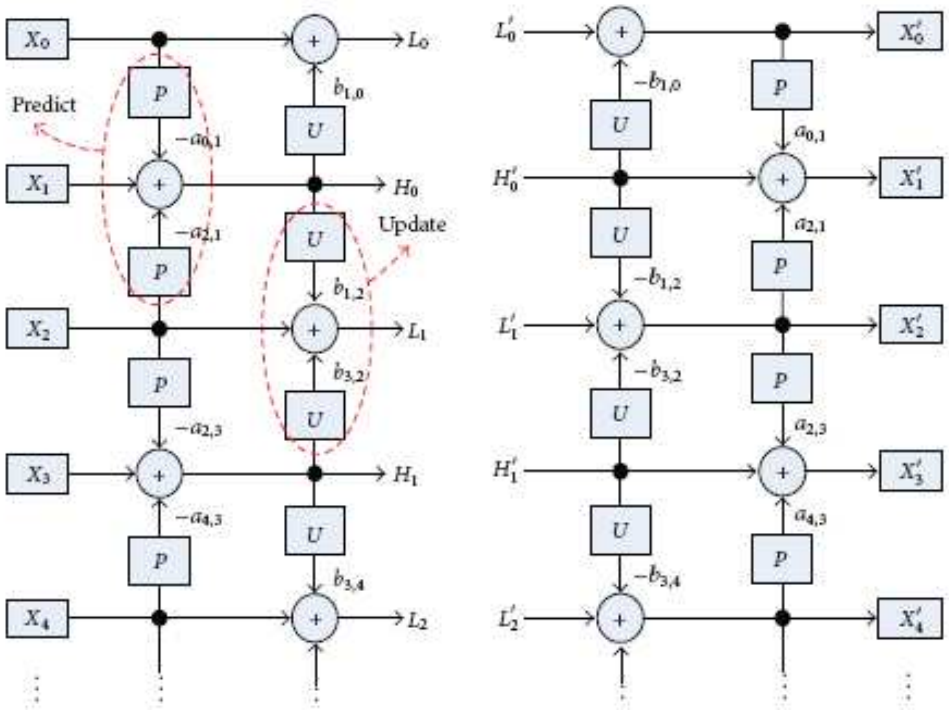
La figura 5.3 mostra il primo livello di decomposizione in sottobande della



(a) Codifica (b) Decodifica

Figura 5.3: Primo livello di decomposizione in sottobande con wavelet lifting Haar

DCWL tipo Haar e la rispettiva decofica, mentre la figura 5.4 è relativa alla DCWL tipo 5/3.



(a) Codifica (b) Decodifica

Figura 5.4: Primo livello di decomposizione in sottobande con wavelet lifting 5/3

Capitolo 6

Risultati sperimentali

In questo capitolo verranno discussi e messi a confronto i risultati ottenuti dai due tipi di algoritmi precedentemente illustrati. Il confronto viene fatto considerando come parametro di qualità il PSNR medio delle immagini ricostruite. In entrambi i casi è stata usata la stessa applicazione, il codificatore KAKADU, per comprimere i coefficienti della trasformata wavelet nella codifica JPEG2000.

6.1 Datasets di test

I due algoritmi proposti sono stati testati sostanzialmente su due set di immagini multiview formate da 8 viste ciascuna e con le camere disposte in modo pressoché lineare lungo un arco di circonferenza. Il sistema implementato è comunque generale e può essere eseguito con un numero arbitrario di viste. I due algoritmi inoltre sono stati implementati considerando solo la componente di luminanza di ogni immagine. L'estensione agli altri canali può essere ottenuta semplicemente riapplicando gli algoritmi per ogni singolo canale.

Il primo set di immagini usato è stato generato da un modello-3D, sintetico, di una cucina (dataset "Kitchen" [1]). La distribuzione delle camere

[1]Dataset, parametri camere, e tutte le informazioni sono disponibili online all'indirizzo

all'interno di questo modello è riportato in Figura 6.1 mentre in figura 6.2

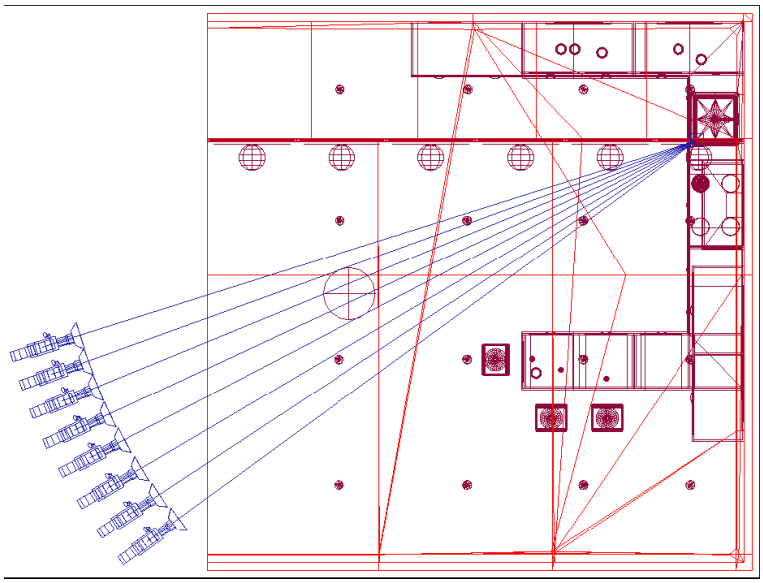
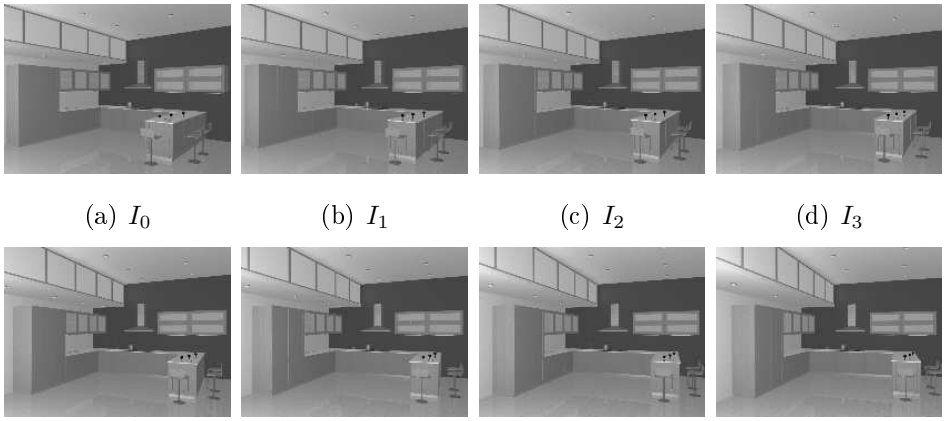


Figura 6.1: Distribuzione delle camere del dataset "Kitchen" visto dall'alto. Le camere vengono numerate da 0 a 7 dal basso.

vengono riportate tutte le 8 viste del dataset. Un'osservazione importante da fare è che questo modello è un modello sintetizzato, quindi non presenta quegli effetti dovuti al rumore, alla distorsioni delle camere e a tutte quelle caratteristiche presenti in una sequenze multiview reale. Questo dataset permette quindi di testare i sistemi di codifica in un caso, per così dire, ideale. Inoltre, avendo a disposizione una mappa di profondità quasi perfetta, tutte quelle problematiche legate al warping, soprattutto nel secondo algoritmo, vengono meno. Le mappe di profondità, infatti, per questo modello sono a 16 bit quindi molto più dettagliate e precise.

Al fine di verificare le prestazioni dei sistemi di codifica proposti con dati più realistici, si è fatto uso della sequenza video multiview "breakdancers"[3]. Per testare gli algoritmi sono stati presi in esame solo le 8 viste del primo fotogramma (figura 6.3). Da notare è che le informazioni di profondità di questo modello, vengono calcolate tramite algoritmi di visione computazionale dal-

http://lttm.dei.unipd.it/downloads/kitchen

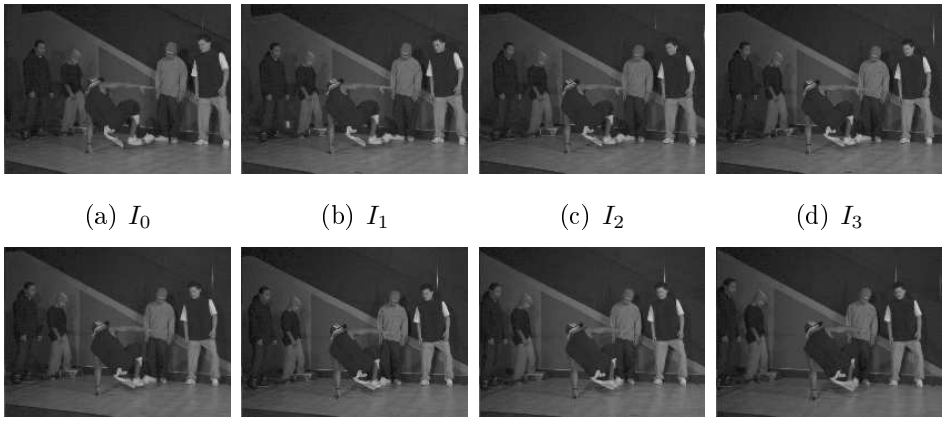


(a) I_0 (b) I_1 (c) I_2 (d) I_3

(e) I_4 (f) I_5 (g) I_6 (h) I_7

Figura 6.2: Dataset "Kitchen".

lo stream video. La precisione è quindi inevitabilmente peggiore rispetto a quella ottenibile direttamente dai modelli sintetici. Inoltre, le mappe di profondità disponibili per questo modello sono a 8 bit quindi ulteriormente di qualità inferiore.



(a) I_0 (b) I_1 (c) I_2 (d) I_3

(e) I_4 (f) I_5 (g) I_6 (h) I_7

Figura 6.3: Dataset "Breakdancers".

Come discusso nella sezione 2.4.1, i valori relativi alle mappe di profondità possono essere codificati attraverso convenzioni diverse. Nei datasets presi in esame vengono utilizzate due tipi differenti di mappatura: il dataset

"breakdancers" adotta la funzione definita dall' equazione (2.18), mentre il dataset "kitchen" adotta quella definita dall'equazione (2.19).

6.2 Risultati

L'utilizzo della trasformata wavelet negli algoritmi proposti nei capitoli 4 e 5 consente, sostanzialmente, di pre-elaborare un dataset di N immagini multiview in modo da ottenerne uno nuovo, sempre di N immagini, tale per cui l'applicazione di una qualsiasi codifica di compressione risulti particolarmente efficiente. Nel nostro caso si è fatto uso, come già affermato, della codifica di compressione JPEG2000, illustrata nel paragrafo 3.2.1, la quale fa uso anch'essa di trasformata wavelet.

6.2.1 Parametri KAKADU

"*KAKADU*"[2] è il software utilizzato per processare i coefficienti wavelet e ottenere la codifica JPEG2000. Indipendentemente dall'algoritmo utilizzato i coefficienti wavelet ottenuti devono essere rappresentati da immagini a 16 bit le quali vengono passate come ingressi al codificatore.

I parametri impostati nelle simulazioni sono riportati in tabella 6.1. Attraverso il parametro *slope*, il quale rappresenta la pendenza della curva rate distorsione, viene impostata la *qualità* della codifica e di conseguenza il rate. Questo parametro è variabile nel senso che viene impostato dell'utente al momento della compressione. Il fatto di impostare la qualità e non il rate desiderato consente di poter comprimere i coefficienti in alta frequenza a rate inferiori mantendendo comunque una qualità uniforme su tutto il dataset delle immagini.

Gli altri parametri che vengono settati sono legati al tipo di dato in uscita dalla pre-elaborazione wavelet. Come detto sopra, dopo la trasformata wavelet tra le varie viste, i dati ottenuti sono degli interi a 16 bit (*Sprecision=16*) con segno (*Ssigned=yes*). Siccome stiamo lavorando su una singola

[2] http://www.kakadusoftware.com

Parametri	Value
slope	*Variabile*
Scomponents	1
Ssigned	yes
Sprecision	16
Qstep	$1/2^{16}$

Tabella 6.1: Parametri KAKADU.

componente *Scomponents=1* mentre Qstep, che rappresenta il passo di quantizzazione, deve essere impostato $1/2^{16}$ in quanto si sta lavorando con parole di 16 bit dove l'informazione, soprattutto per le alte frequenze, sta nei bit meno significativi.

6.2.2 Grafici

I grafici in figura 6.4 e figura 6.5 rappresentano le curve del PSNR in funzione del bit rate nei due tipi di algoritmi implementati. Questi grafici fanno riferimento alla sequenza "Kitchen", e sono stati realizzati rispettivamente con la trasformata wavelet HAAR (figura 6.4) e la trasformata wavelet 5/3 (figura 6.5). In entrambe le figure, inoltre, è presente un ulteriore curva (*KAKADU*) ottenuta semplicemente applicando la codifica JPEG2000 alle 8 viste del dataset senza nessuna pre-elaborazione.
I grafici 6.6 e 6.7 invece fanno riferimento alla sequenza "Breakdancers" e sono stati ottenuti con gli stessi criteri descritti per la sequenza "Kitchen": la figura 6.6 mostra il grafico del PSNR con trasformata wavelet HAAR mentre la figura 6.5 con trasformata wavelet 5/3 nei tre casi implementati.

Questi grafici mostrano chiaramente come il primo algoritmo sia molto più performante: in entrambe le sequenze infatti la curva relativa al primo algoritmo sta molto al di sopra rispetto a quella del secondo, mentre rispetto alla curva "KAKADU" la curva sta sopra solo a bassi bit rate. Un esempio del risultato dell'algortimo è mostrato in figura 6.8. Pur avendo prestazioni

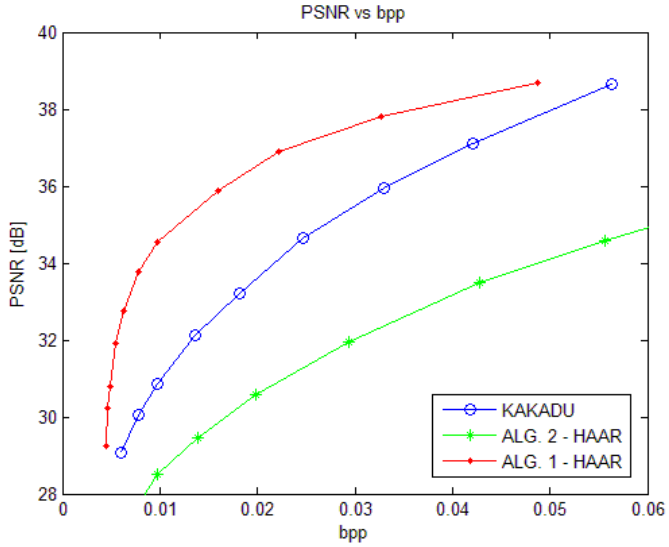


Figura 6.4: Confronto tra le prestazioni i due algoritmi con trasfrormata wavelet HAAR sul dataset "kitchen"

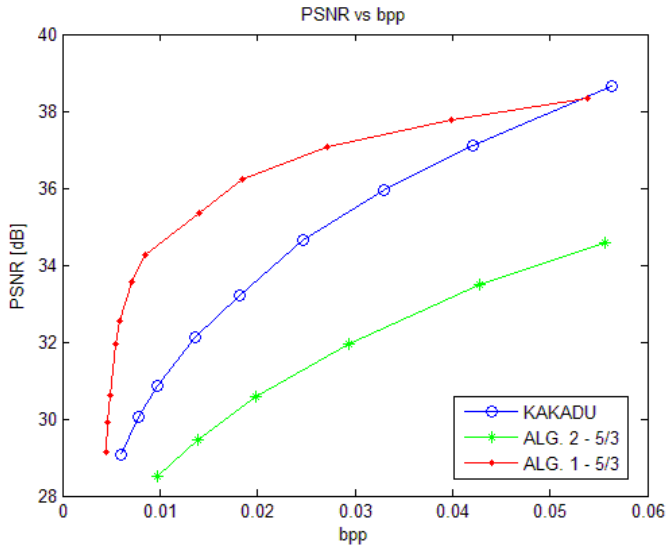


Figura 6.5: Confronto tra le prestazioni i due algoritmi con trasfrormata wavelet 5/3 sul dataset "kitchen"

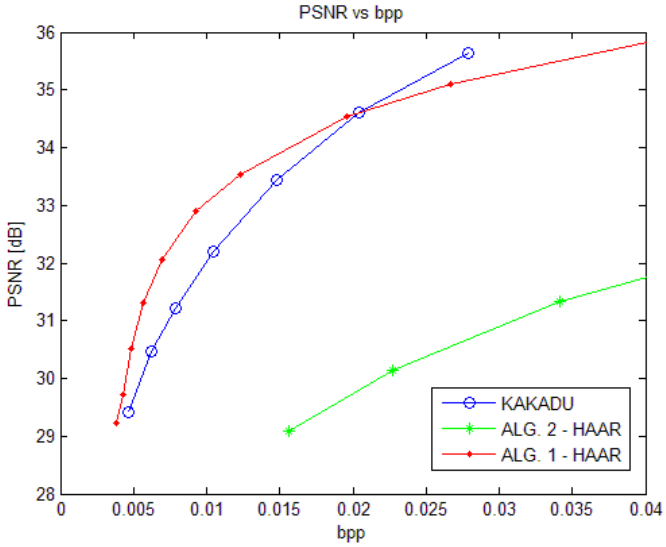


Figura 6.6: Confronto tra le prestazioni i due algoritmi con trasfrormata wavelet HAAR sul dataset "Breakdancers"

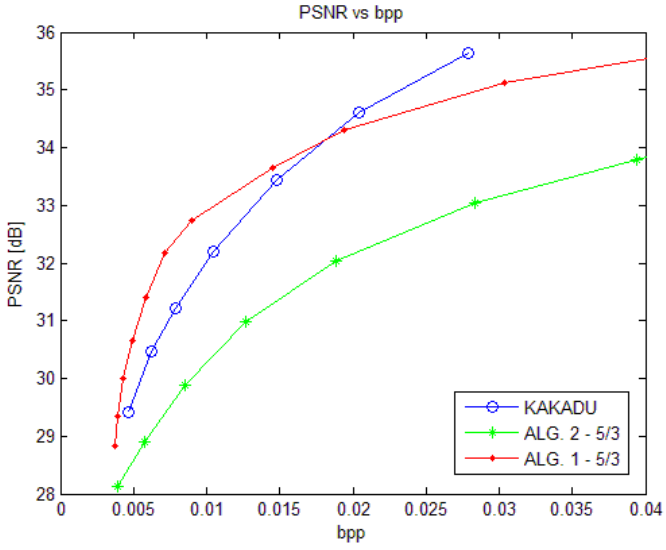


Figura 6.7: Confronto tra le prestazioni i due algoritmi con trasfrormata wavelet 5/3 sul dataset "Breakdancers"

migliori, è interessante osservare come le curve relative all'algoritmo 1, per entrambe le sequenze e tipologia di trasformata wavelet applicata, tendano a stabilizzarsi. Questo fatto è dovuto principalmente alle operazioni di warping che vengono eseguite prima del wavelet lifting.

(a) Originale (b) Algoritmo 1 a 0.016 ppb

Figura 6.8: Prima immagine della sequenza "kitchen"

L'applicazione della trasformata wavelet prima di compattare le immagini risulta di vitale interesse in quanto a parità di bit rate la qualità finale è decisamente migliore. La figura 6.9, riguardante l'algoritmo 1 e sequenza "kitchen", mostra chiaramente i benefici della trasformata, in questo caso HAAR. La curva verde infatti è stata ottenuta semplicemente escludendo il passaggio relativo al wavelet lifting: in pratica questa curva corrisponde alla compressione diretta dell'image-stack. Come si evince chiaramente dal grafico il solo warping induce a un decadimento delle prestazioni se confrontato con la curva KAKADU (che appunto corrisponde alla semplice compressione delle immagini). Con la pre-elaborazione wavelet invece i coefficienti che si ottengono risultano particolarmente efficienti in fase di compressione ottenendo in questo modo, a parità di PSNR, un bit-rate decisamente inferiore.

Un fatto interessante è che tale aumento di efficienza nel secondo algoritmo non si verifica. Il problema evidentemente sta nell'utilizzo dei parametri di compressione passati al KAKADU. I coefficienti ottenuti dal primo al-

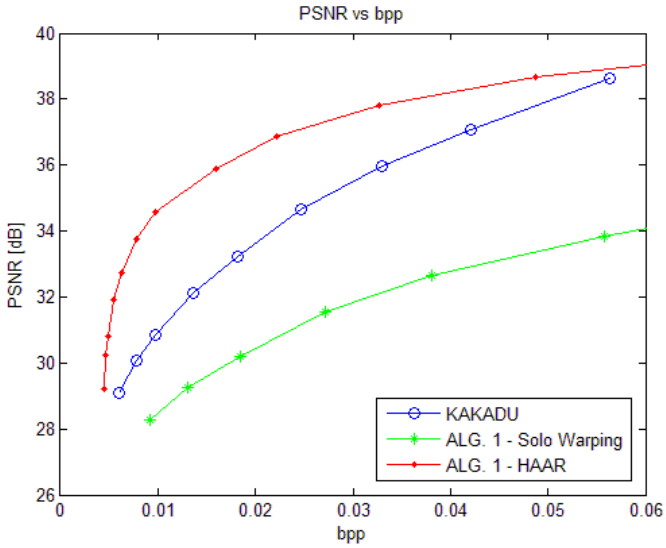


Figura 6.9: Confronto tra trasfrormata wavelet HAAR e applicazione solo del warping nell'algoritmo 1 con dataset "kitchen"

goritmo, infatti, presentano alle alte frequenze zone contenenti informazioni legate alle occlusioni tra le viste. Molto probabilmente applicando in modo mirato la compressione con rate diversi in funzione del tipo di zona, l'efficienza aumenterebbe, purtroppo però questo non è possibile con la versione del software KAKADU utilizzata. Il software di compressione è infatti pensato per comprimere immagini mentre qui si è cercato di adattarlo alle sottobande prodotte dal lifting, probabilmente una codifca ad-hoc dei coefficienti delle sottobande darebbe migliori risultati.

Nelle figure 6.10, 6.11, 6.12, 6.13 viene messo a confronto l'efficienza nell'utilizzare la trasformata wavelet HAAR o 5/3 nei due algoritmi. I grafici 6.10 e 6.11 mostrano che per quanto riguarda il primo algoritmo le due trasformate, soprattutto a bassi bit rate, sono quasi coincidenti, mentre all'aumentare del bit rate la trasformata HAAR risulta leggermente più efficiente. Per quanto riguarda il secondo algoritmo invece le curve relative alla trasformata 5/3 risultano migliori rispetto a quelle della trasformata HAAR. In figura 6.12 e 6.13 vengono riportate tali curve rispettivamente per la se-

quenza "kitchen" e "Breakdancers" e si può osservare chiaramente come a parità di bit rate il PSNR sia decisamente maggiore.

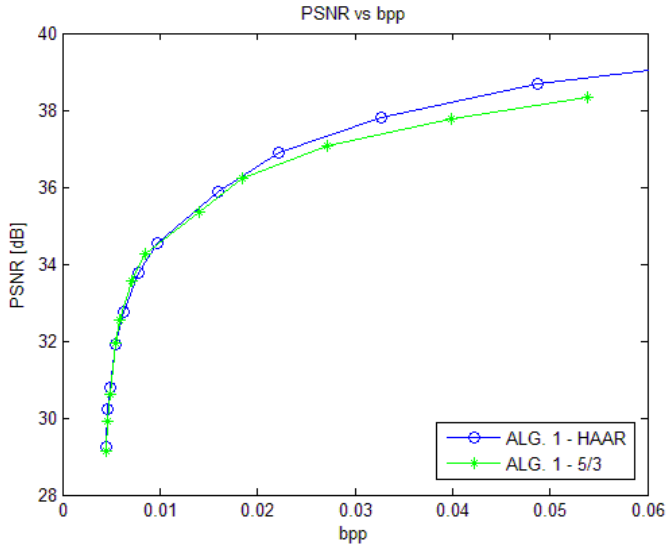


Figura 6.10: Confronto tra trasfrormata wavelet HAAR e 5/3 con algoritmo 1 e dataset "kitchen"

Le immagini 6.14, 6.15, 6.16, 6.17 rappresentano la suddivisione del bitstream totale nei due algortimi implementati. Tali grafici fanno riferimento alla sequenza "kitchen" codificata con i due algoritmi sia con trasfromata HAAR (figure 6.14 e 6.16) che con traformata 5/3 (figure 6.15 e 6.17).

Il bitstream totale può essere quindi scomposto al più in 3 sotto flussi: il *bitstream bassa frequenza* corrisponde alla prima sottobanda con le informazioni in bassa frequenza, il *bitstream alta frequenza* invece comprende tutte le altre sottobande in alta frequenza mentre il *bitstream occlusioni* corrisponde all'immagine formata da blocchetti 8×8 la quale viene compressa con JPEG (ovviamente il primo algoritmo non produce nessun stream occlusioni).

Questi istogrammi mostrano chiaramente, come già illustrato precedentemente, che l'alta frequenza dell'algoritmo 2, la quale contiene anche tutte quelle informazioni relative alle occlusioni, non sia possibile comprimerla tan-

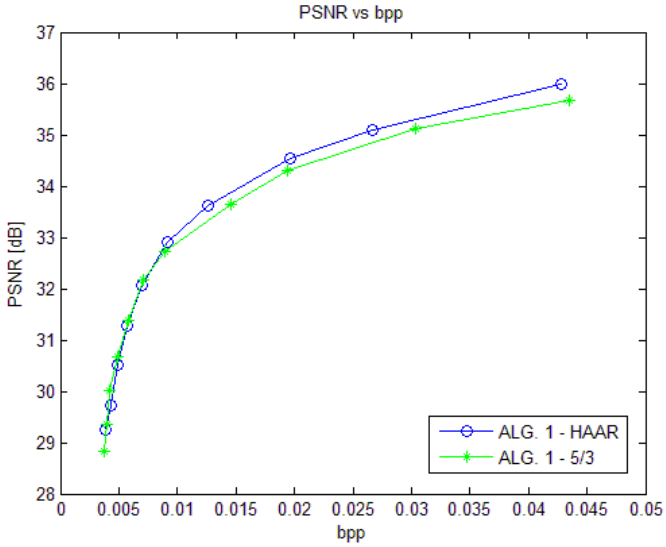


Figura 6.11: Confronto tra trasfrormata wavelet HAAR e 5/3 con algoritmo 1 e dataset "Breakdancers"

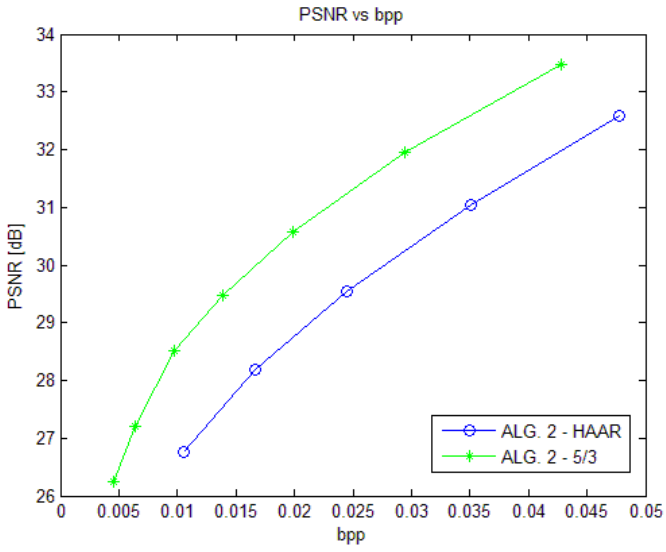


Figura 6.12: Confronto tra trasfrormata wavelet HAAR e 5/3 con algoritmo 2 e dataset "kitchen"

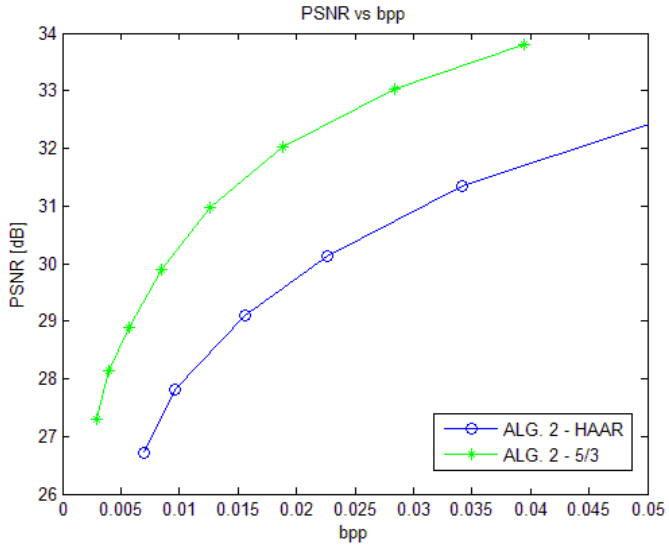


Figura 6.13: Confronto tra trasfrormata wavelet HAAR e 5/3 con algoritmo 2 e dataset "Breakdancers"

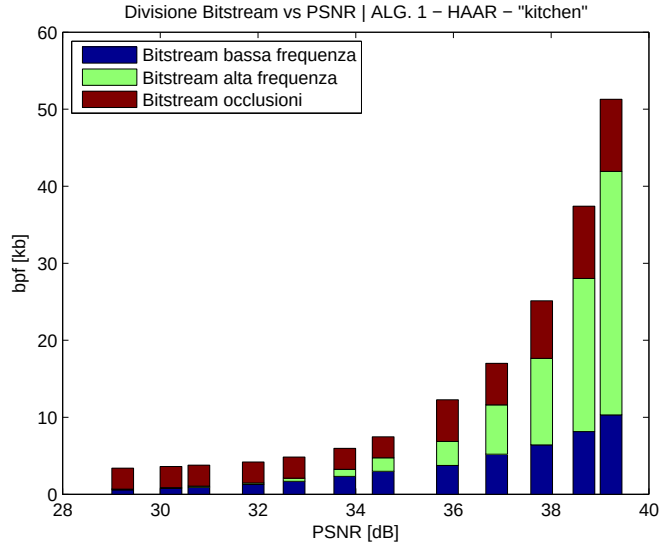


Figura 6.14: Divisione Bitstream del dataset "kitchen". Algortimo 1 con wavelet HAAR

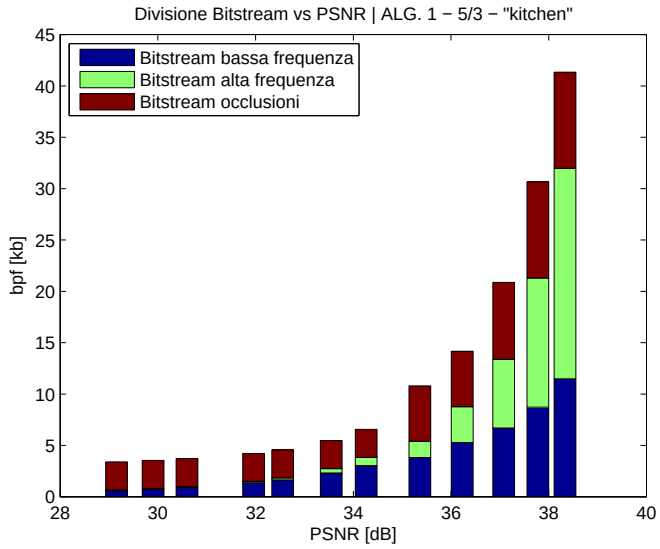


Figura 6.15: Divisione Bitstream del dataset "kitchen". Algortimo 1 con wavelet 5/3

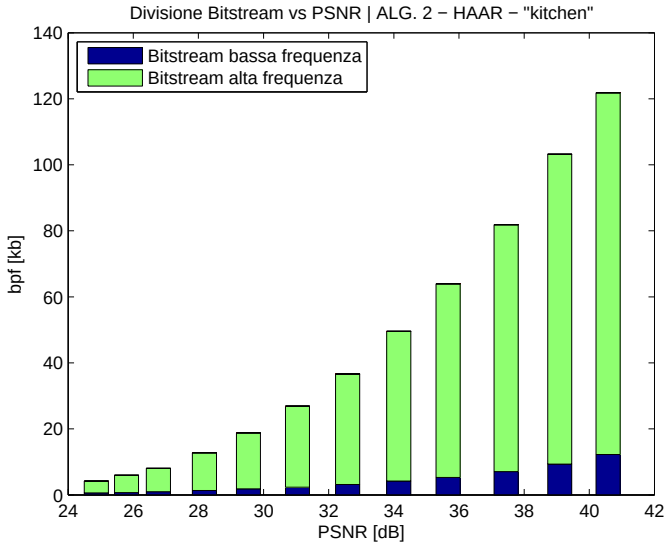


Figura 6.16: Divisione Bitstream del dataset "kitchen". Algortimo 2 con wavelet HAAR

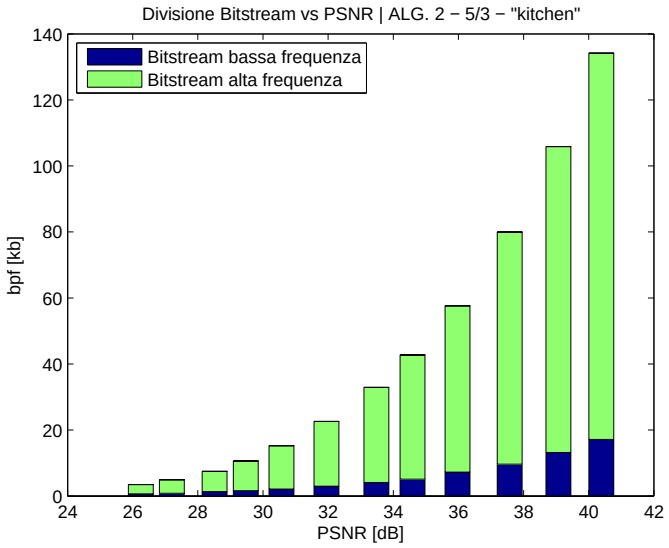


Figura 6.17: Divisione Bitstream del dataset "kitchen". Algortimo 2 con wavelet 5/3

to quanto il secondo algoritmo a parità di slope con JPEG2000. Inoltre è possibile osservare come l'algoritmo 1, a bassi bit rate, nonostante funzioni meglio, sia comunque molto penalizzato dal bitstraem delle occlusioni il quale, per mantenere la qualità richiesta, occupa in proporzione molto più spazio dei coefficienti dalla wavelet.

Capitolo 7

Conclusioni

In questa tesi sono stati implementati e messi a confronto due schemi di codifica di immagini multiview che fanno uso di trasformata wavelet tra le varie viste e successivamente comprimono con JPEG2000 i coefficienti ottenuti. In entrambi gli algoritmi viene quindi applicata una pre-elaborazione mediante uno schema di wavelet lifting prima di passare alla compressione vera e propria. La differenza sostanziale tra i due approcci sta nell'eseguire le operazioni di warping: all'interno della wavelet tra le viste adiacenti per quanto riguarda lo schema 1 (capitolo 5), o a priori della trasformata wavelet proiettando in un'unica vista e gestendo le occlusioni in modo indipendente per quanto riguarda lo schema 2 (capitolo 4).

I dati sperimentali mostrano come il primo approccio sia decisamente migliore rispetto al secondo, a parità di parametri utilizzati per la compressione JPEG2000. Tale metodo inoltre si è rivelato ottimale a bassi bit-rate evidenziando come l'uso della trasformata wavelet consenta di ottenere coefficienti in alta frequenza che, a parità di slope, possono essere compressi molto più efficacemente. Le prestazioni a bit rate più elevati invece, dipendono molto dall'accuratezza dei dati geometrici. Le curve del secondo algoritmo tendono infatti a stabilizzarsi e si è potuto constatare che le sequenze sintetizzate, a parità di bit-rate consentono di ottenere PSNR più elevati, grazie

soprattutto alla migliore precisione della depth map.

Dai dati sperimentali inoltre si è potuto osservare come i coefficienti delle immagini ad alta frequenza ottenuti dalla DCWL dell' algoritmo 2, non risultano essere ottimizzati per una compressione JPEG2000. Il bit rate che si ottiene per ogni immagine ad alta frequenza, infatti, è molto vicino a quello dell'immagine rappresentante il coefficienti in bassa frequenza, a parità di slope. Il decadimento delle prestazioni è quindi legato alla compressione JPEG2000 più che alla trasformata wavelet tra le viste. Il problema potrebbe risiedere nell'informazione relativa alle zone occluse, tra le viste adiacenti, che emerge nelle immagini in alta frequenza. Un rimedio potrebbe quindi sussistere nell'identificare queste zone e comprimerle con rate diverso dal resto dell'immagine in modo da diminuire il bitstream totale.

L'utilizzo della trasformata wavelet è di sicuro uno strumento molto importate ai fini di una codifica efficiente di immagini multiview, come dimostrano i dati ottenuti per l'algoritmo 1, sopratutto a bassi bit rate. D' altro canto dall'algoritmo 2 si può osservare come una codifica di compressione non idonea dei coefficienti porti al decadimento delle prestazioni.

Bibliografia

[1] M. Zamarin, S. Milani, P. Zanuttigh, and G. M. Cortelazzo, "A Novel Multi-View Image Coding Scheme based on View-Warping and 3D-DCT," *Elsevier JVCI - Special Issue on Multi-Camera Imaging, Coding and Innovative Display: Techniques and Systems,* Feb. 2010.

[2] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *Image Processing, IEEE Transactions on,* vol. 13, pp. 600-612, Apr. 2004.

[3] L. C. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," *ACM Trans. Graph.,* vol. 23, no. 3, pp. 600-608, 2004.

[4] A. Fusiello, "Visione computazionale. Appunti delle lezioni." *http://ilmiolibro.kataweb.it/schedalibro.asp?id=229488.* Jun. 2008.

[5] R. I. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision." Cambridge University Press, second ed., 2004.

[6] Y. Sheng, "WAVELET TRANSFORM." *The transforms and applications handbook.* Ed. by A. D. Poularikas. P. 747-827. Boca Raton, Fl (USA): CRC Press, 1996. The Electrical Engineering Handbook Series.

[7] C. Valens, "A Realy Friendly Guide to Wavelets" *http://perso.wanadoo.fr/polyvalens/clemens/wavelets/wavelets.html* 2004

[8] Daubechies, "TEN LECTURES ON WAVELETS. 2nd ed." *CBMS-NSF regional conference series in applied mathematics 61.* Philadelphia: SIAM, 1992.

[9] Burrus, C. S. and R. A. Gopinath, H. Guo. "INTRODUCTION TO WAVELETS AND WAVELET TRANSFORMS, A PRIMER." Upper Saddle River, NJ (USA): Prentice Hall, 1998.

[10] Mallat, S. G. A THEORY FOR MULTIRESOLUTION SIGNAL DECOMPOSITION: THE WAVELET REPRESENTATION. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 7 (1989), p. 674- 693.

[11] C.-L. Chang, X. Zhu, P. Ramanathan, and B. Girod, "Interview wavelet compression of light fields with disparitycompensated lifting", *Visual Communications and Image Processing*, vol. 5150 of Proceedings of SPIE, pp. 694Ű706, Lugano, Switzerland, July 2003.

[12] B. Girod, C.-L. Chang, P. Ramanathan, and X. Zhu, "Light field compression using disparity-compensated lifting," *Proceedings of the IEEE International Conference on Acoustics*, Speech and Signal Processing (ICASSP Š03), vol. 4, pp. 760-763, April 2003.

[13] P. Lasang and W. Kumwilaisak, "Rate Distortion Analysis and Bit Allocation Scheme for Wavelet Lifting-Based Multiview Image Coding"

[14] A. Secker and D. Taubman, "Motion-compensated highly scalable video compression using an adaptive 3D wavelet transform based on lifting," in Proceedings of the IEEE International Conference on Image Processing (ICIP '01), vol. 2, pp. 1029-1032, October 2001.

[15] B. Pesquet-Popescu and V. Bottreau, "Three-dimensional lifting schemes for motion compensated video compression," in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal

[16] N.Mehrseresht and D. Taubman, "Adaptively weighted update steps in-motion compensated lifting based scalable video compression," in Proceedings of the IEEE International Conference on Image Processing (ICIP '03), vol. 3, pp. 771-774, September 2003.

[17] Khalid Sayood, "Introduction to Data Compression, Third Edition " Third edition

[18] J. W. Woods, Multidimensional Signal, Image and Video Processing and Coding. Academic Press, 2006.

[19] D. taubman. - Directioanlty and scalability in image and video compression. Ph.D. thesis, University of california at berkeley, May 1994

[20] D. taubman and A.Zakhor. - Multirate 3-D subband coding with Motion Compensation- University of california at berkeley, settembre 1994